Department of Computer Science and Software Engineering
University of Canterbury

May 2017

Calibration using Supervised Learning for Low-Cost Air Quality Sensors

A thesis submitted in partial fulfilment of the requirements
for the Degree of Master Science in Computer Science
in the University of Canterbury by

Boney Bun

Supervisor : Associate Professor Dr. Andreas Willig
Co-Supervisor : Dr. Malcolm Campbell

Abstract

Low-cost environmental sensors encounter challenges of reliability and accuracy. This thesis aims at increasing the readings accuracy of low-cost air quality sensors, particularly using three Air Quality Egg (AQE) version 2 to measure the CO and NO_2 concentrations in the air, in a supervised manner, by proposing the outlier and adjustment modules. The outlier module detects and eliminates noise in the sensor readings, while the adjustment module aims to increase the sensors' reading accuracy. Four proposed detection schemes and three mathematical algorithms were trained and tested in the outlier module and adjustment module, respectively.

The temperature, humidity, and CO sensors on the AQEs had good readings agreement based on the index of agreement (*d*-value), except for the NO_2. Thus, the selection of the schemes for the outlier module can be based on the sensors' characteristic. The scheme that verifies the adjacent nodes before marking an outlier has better ability to classify the outlier than a scheme that does not. Artificial neural network (ANN) outperformed the other two mathematical techniques in the adjustment module because the accuracy performance is influenced by the sensors readings agreement and the performance of the outlier module.

The CO, temperature, and humidity sensors on the three AQEs can fit the CO reference from Environment Canterbury's data by at least 80% with or without the existence of the outlier module. On the other hand, the NO_2, temperature and humidity sensors can fit the NO_2 reference by 40% and at least 80%, respectively, without and with the filter from the outlier module.

Acknowledgements

Although seems to be old fashion way, but only God makes this thesis (and my study) possible. HE would definitely be the first on the thankful list.

Special thanks to NZ Scholarship and UC's Student Care Advisors for allowing me having a life time experience.

My wife and son who has supported me through all the distances and exhaustive time.

I would like to express thanks to my supervisor, Andreas Willig, for his time, critical thinking, and valuable feedback during the thesis. I would also express thanks to Malcolm Campbell, for the feedback and encouragement.

Many thanks to numerous friends and colleagues whom cheering my life through ups and downs during the life at New Zealand. Members of Ha and PPIC group (in no particular order): Gerry, Emir, Frans, Danny, Rendy, Kat. I have learnt a lot of from you guys. 343 lab mates whom supported in silence giving a chance having a conducive working space.

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1: Introduction

Scientific evidence has shown that air pollution contributes to respiratory diseases, as reported by the World Health Organization (WHO) [1]. According to this report, children and the elderly are more sensitive to air pollution. Therefore, real-time ambient air quality information could be an important way to protect sensitive citizens, if it were made accessible online by the relevant authorities, using instruments to monitor air quality. Currently, monitoring instruments provide some air quality information, but usually only covering broad areas of a city. Yet, the concentration of air pollutants is likely to vary locally and be location-dependent, such as in industrial areas, busy roads, around houses burning firewood, near forest fires [2], and so on. Current instruments do not give such detailed information on small specific areas in ways that could potentially be valuable to residents in those areas.

There is a foreseeable trend of spatial temporal monitoring in the future – the 'Internet of Things'. The trend is for a growing number of commercial sensing devices, as advanced low-cost environmental sensors become more available and affordable. There is a further trend towards increased awareness regarding healthy lifestyle, as people become eager to assess their surroundings for health factors such as air pollution. These two factors may encourage the public in using low cost sensor products as part of their daily activities. As a result, there may be a plethora of spatial-temporal data in the future. Furthermore, these trends may lead to the use of opportunistic sensing concepts [3] and the development of smart cities using these concepts.

We are interested in the Air Quality Egg (AQE) in this study. The AQE is an air quality monitoring sensor that enables online real-time monitoring at specific locations. The AQE

project received initial funding from a crowd-funding website and is an open source product. AQE has a growing number of users around the world [4]. In comparison to conventional air monitoring equipment, typically costing more than US$ 5,425 (€5,000) [2], an AQE is relatively cheap and affordable for most people, at a cost of about US $240. Consumers can easily connect an AQE to their wireless networks in their homes. Then, ambient air quality information is accessible through the AQE's website or the users can look directly at the device's LCD display. Apart from giving an hourly average on the website, the AQE data receives no other treatment or analysis. Adding a data treatment or an analysis layer to the system can potentially deliver added value to the consumers. Most of the Open Source Systems (OSSs) deliver very robust and reliable projects within an affordable budget even for start-up companies. So, it is likely that OSS tend to attract more people, ranging from amateur to professional users. The AQE is an open source system (OSS) which both amateurs and professionals may use for obtaining ambient air quality information; more advanced users, such in the Air Quality Sense Box project [5], may customize it to suit their specific needs. AQE hardware and software information are available online. Following a successful campaign on Kickstarter to raise initial funding for the AQE project [6], the company now sells version two of AQE on their own site.

There are two drawbacks to the use of low-cost sensors: low accuracy, and spurious data due to location. Firstly, regarding low accuracy, Choi *et al* [7] pointed out that micro-electro-mechanical system (MEMS) sensors are inaccurate when compared to expensive sensors. To overcome this problem, sensor calibration may be needed, in which adjustment is made in a controlled laboratory. However, this solution may not be appropriate or affordable to the public. Thus, in this project we propose an affordable way to improve the accuracy of low-cost sensors by post-processing its output using mathematical algorithms.

There is much literature on the use of various mathematical algorithms in the data analysis stage. For example, two studies of Spinelle *et al* [8, 9] showcased the use of linear regression, multilinear regression, and artificial neural networks. Vito *et al* [10, 11] use a multivariate correlation technique to calibrate the electronic node in their two studies. Therefore, applying mathematical algorithms in the data analysis has a good possibility of fixing the accuracy of the low-cost sensors.

The second challenge might occur from the location or placement of low-cost sensors, particularly outdoors, where the performance of the sensors can be influenced by the ambient environment, such as the temperature, relative humidity, and wind speed. A solution to this might include having redundant sensors on a site, where one AQE can potentially detect anomalies of other AQEs. Therefore, in this project we also explore using redundant low-cost sensors in a site. We use mathematical algorithms as an approach to both problems.

To solve the two challenges mentioned above, we built a software calibration solution specifically for Air Quality Egg version 2. There already exists a current system implementation of this architecture, and we have proposed a modification of this. On the current system implementation, each AQE sends data to the OpenSensors, while data visualisation can be obtained from AQE's website or Xively [12]. In our proposed system architecture, illustrated in Figure 1.1, a processing server is added between the OpenSensors [13] and the end users. The additional processing server (called the 'R server') can pull the sensors' data from the OpenSensors using the MQ Telemetry Transport (MQTT) protocol. MQTT is a machine-to-machine messaging protocol aimed to deliver lightweight publish/subscribe messages [14]. The data can also be parsed, analysed, and recalculated in the R server. A mathematical

module for data processing, called the AQE module, will be located in the server. The AQE module is divided into two:

- **Outlier module**

Air quality data may contain a sudden change and a seasonal pattern [15]. Thus, distinguishing whether a reading in the data is a normal reading or an outlier can be a big challenge, a sudden change in the reading can result from a change in the environment, not the sensors' fault. The outlier module aims to classify and filter whether an hourly sensor reading at a certain time is an outlier or not. All the hourly AQE readings will be filtered by the outlier module first before going to the adjustment module. If an outlier is found in the reading, the reading will not proceed to the adjustment module as it may affect the calibration process.

- **Adjustment module**

The adjustment module is the central part of this thesis and its goal is calibrating the hourly AQE reading. All the readings passing the adjustment module are calculated mathematically. The result is a calibrated output at a certain time.

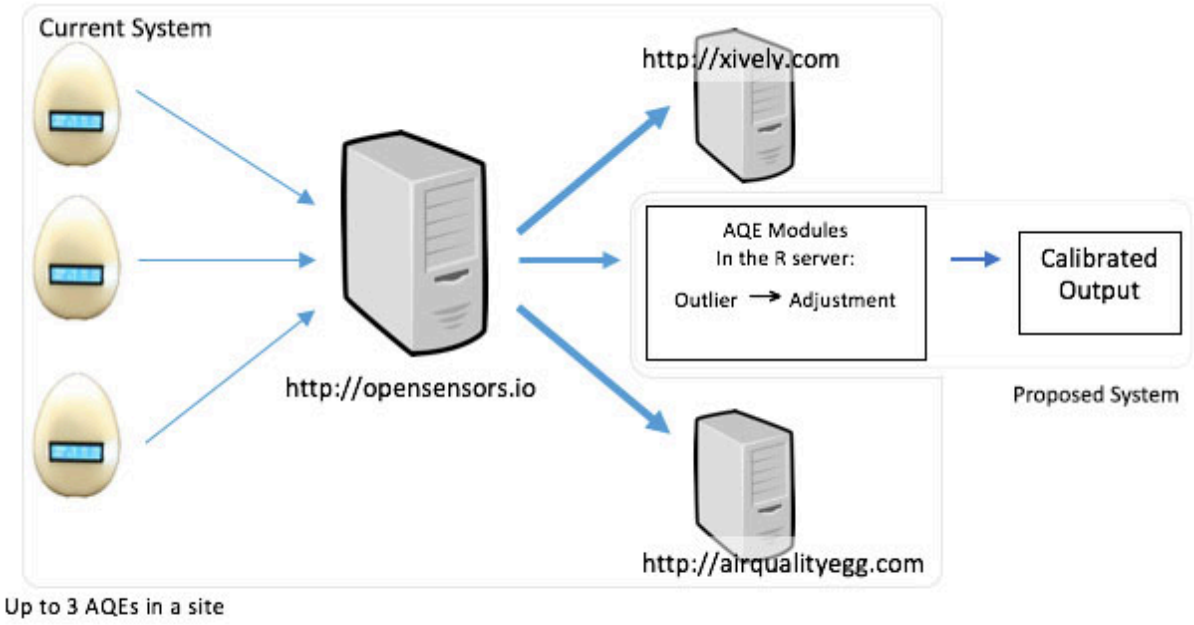


Figure 1.1 Information flow of the proposed system architecture [1]

We apply supervised learning to train the classification algorithm in the outlier module and the calibration algorithm in the adjustment module [16, 17]. Past data is useful to train the classification algorithm in the outlier module. Whereas, a portion of AQE data is reserved as the training data for the calibration algorithm in the adjustment module. The final output of AQE module is calibrated output.

The impact of the outlier module to the adjustment module is assessed by comparing the calibration performance of the adjustment module with or without the presence of the outlier module. Each module is also evaluated. The outlier module is assessed using classification accuracy by counting the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) of the module. TP and TN mean the module has

[1] The pictures are taken from http://www.airqualityegg.com and http://www.cliparfrefor.com

correctly identified the sensor readings as normal and outlier, respectively. FP and FN indicate the module has mistakenly identified the normal and outlier readings, respectively. The classification accuracy percentage of the outlier module is then calculated based on the number of TP, TN, FP, and FN. Meanwhile, the adjustment module is assessed by using statistical methods which are: coefficient of determination (r-square), root mean square error (RMSE), and d value. 100% accuracy for the outlier module and a value of 1 of *R-square* or *d value* for the adjustment module are the ideal result for the two modules.

We present the past studies related to this thesis in Chapter 2. The detail of the experiment for the proposed system is presented in Chapter 3. The root of the proposed system is in the outlier and adjustment modules. Both modules are presented in detail in Chapters 4 and 5 explaining the proposed and evaluation methods, training phase, testing phase and results. Finally, the discussion of the findings and future work are in Chapter 6.

Chapter 2: Related Work

This chapter presents the literatures regarding low-cost environmental sensors. Section 2.1 describes the current monitoring system used by the New Zealand government and the reasons people with health concerns monitor their ambient air quality. On-going alternative developments to the conventional monitoring system are presented in Section 2.2. Section 2.3 and Section 2.4 explain artificial neural networks and anomaly detection techniques found in the literature.

2.1 Monitoring Equipment

Scientific evidence suggests that harmful pollutants have a significant impact on human health [18]. Different standards have been laid down to minimize the impact of these harmful pollutants to people; of particular interest to us is an emission performance standard. Most countries have institutions that set these standards; Environment Canada (Canada), the Environmental Protection Agency (the USA), the European Union (Europe), Environment Canterbury (Canterbury, New Zealand), and China's State Environmental Protection Administration (SEPA) are examples. These standards require constant monitoring systems in order to be effective and ensure compliance by all stakeholders. Such monitoring equipment is the subject of review next.

To detect the presence of a particular gas accurately, Fourier transform infrared (FTIR) instruments, gas chromatographs, and mass spectrometers are commonly being deployed by authorities for long-term monitoring systems [19]. Monitoring and assessment may cover a partial or a whole area. The monitoring systems are not only useful for standards compliance;

they can also be useful to environmental policy makers in order to take action regarding environmental issues. Furthermore, the monitoring of data can be used as an evaluation tool to measure the effectiveness of a policy.

Due to its accuracy, operating a sophisticated monitoring system requires a lot of resources. The system demands skilful operators, with a high maintenance cost, huge space, and large budget [19]. Johnson [20] described the authority's annual budget spent on a water monitoring system through the implementation of the 1972 Clean Water Act (CWA). Less than half of the $982 million from government expenses went to water-pollution monitoring in 2001 according to Johnson's report. Lovet *et al* [21] gave a good reason to keep using this expensive system, they argued that although the monitoring programs need extensive resources, the cost of monitoring programs is cheaper than the cost of policy implementation. They noted that the cost of monitoring and implementation might not have been measurable in the past due to the lack of data.

2.1.1 Authority Monitoring

The Institute of Environmental Science and Research (ESR), and the National Institute for Water and Atmosphere (NIWA) are among New Zealand's institutions monitoring air pollution levels. The Resource Management Act 1991 has given rights to the Regional Council to control air pollution in New Zealand. Any activities releasing air pollutants require a consent from New Zealand's regional offices and local authorities.

According to the New Zealand Institute of Chemistry (NZIC), the country monitors two types of pollutants; primary, and secondary pollutants [22]. Primary pollutants are particulates (smoke, dust, and haze), sulphur dioxide, carbon monoxide, oxides of nitrogen,

benzene, hydrogen sulphide, and fluorides. A secondary pollutant is any substance derived from two or more chemicals, for example, the sulfuric acid of acid rain, which is a mix of sulphur dioxide, water, and air.

The methods for detecting air pollution have been used for more than 30 years. There are two types of monitoring methods; manual, and instrumental methods [22]. Some examples of manual methods are; passive samplers, paper tape samplers, bubbler systems, and dust deposition. Examples of instrumental methods are; non-dispersive infra-red (NDIR), chemiluminescence, flame photometric analysers, fluorescence monitors, and suspended particulate monitoring methods. Manual detection methods demand intensive labour and provide little information, so have progressively been replaced by instrumental methods [22].

2.1.2 Citizen Monitoring

Apart from the government, ordinary people are apparently eager to monitor their environment. Entrikin [23] proposed that a sense of attachment to a place can be a driving factor. Korten [24] has a similar suggestion, recommending that a sustainable population requires the establishment of attachment to its community and environment. Furthermore, Kruger and Shannon [25] proposed that civic social assessment could help people in the community to understand themselves and their environment.

A group of people interested to assess their surroundings can sometimes be referred as citizen scientists, in an activity of community science. They have various backgrounds (public, authorities, industries, academics, and local institutions) and various levels of expertise (non-professional and professional scientists). These people group themselves together based on having something in common. Kerr *et al* [26] reported that 8,000

volunteers were recorded and involved in more than 338 various environmental monitoring activity programs in the USA from 1988 to 1992. Conrad and Daoust [27] conducted surveys and interviewed all Community Based Monitoring (CBMs) groups in the Province of Nova Scotia, Canada in order to understand the advantage of CBM. The Waterkeeper Alliance was an example of global cooperation among CBMs, working to protect ecosystem and water quality in 15 countries, including the USA, Australia, India, Canada, and the Russian Federation [28].

The motives of the citizen science movement may be driven by environmental concerns from the above examples. Uncertainty and lack of funding in government monitoring are potential explanatory factors for citizen monitoring; an example can be found in the study of Savan *et al* [29]. The Canadian Citizens' Environment Watch was founded by three university professors in 1996 as a response to a substantial decrease in funding provided by the Ministry of Environment and Energy between 1995 and 1998. The organisation had independently monitored, and provided education and supervision to the local environment.

2.2 Low Cost Sensors

Many findings strengthen a correlation between air pollution and health. For example, Douglas *et al* [30] studied the air pollution effect on human health by eliminating the effect of smoking and other health risk factors. They found that air pollution does contribute to mortality in six US cities. Another APHEA2 Project study [31], conducted in 29 European cities, found that ambient particles are most likely affecting the mortality rate. The study claimed a relation between the daily number of lives lost and the day-to-day changes of PM10, black

smoke, and NO_2 concentrations. However, the effect of carbon monoxide to human health may not be directly clear, as pinpointed by Alan *et al* in 2002 [32], whose study suggests that more data is required to gather stronger evidence regarding the relationship between carbon monoxide and human health. Epidemiological studies may find ways to use spatial-temporal monitoring for gathering long-term data in an effort to understand patterns, causes, and effects of health conditions and disease in certain populations. These studies may need various types of sensors covering a specific area in order to enable fine-grained analysis.

It is possible that the need of spatial-temporal monitoring may trigger the use of low-cost sensors in the future. The reliability of spatial-temporal monitoring becomes possible with advancing technology in the fabrication of sensors, particularly MEMS. The advance of MEMS technology has driven many sensor developments and subsequent industrial applications [33]. Types of MEMS sensors, such as pressure sensors, microphones, and accelerometers, are known to be reliable and widely used in many industries. Meanwhile, the development of gas sensors is still underway to achieve the same level of reliability as other type of sensor because MEMS technology cannot be fully implemented into gas sensors [33].

Because MEMS technology has not been fully usable in gas sensors, the performance of materials for gaseous low-cost sensors has been a subject of research. Instead of using silicon in the production of metal-oxide semiconductor (MOX) sensors, Vasiliev *et al* [34] added ceramic MEMS technology.

Looking at the drawbacks of gaseous low-cost sensors compared to sophisticated analysis instruments, the output quality of low-cost sensors has not met the requirement of any formal standard, such as the Air Quality Directive 2008/50/EC [35]. Hence, it is likely to be less accurate, yet low-cost sensors may still be usable to those with a community science

interest. Castell *et al* [2] point out there may be other criteria these sensors can meet in the future.

A way to overcome the lack of accuracy problem is to conduct on-field calibration, for example, in detecting benzene [10], CO and NOx [11]. Another calibration approach is to use artificial mixed gas to help low-cost sensors distinguish targeted gas from other gases [36, 37]. Two MiCS-5521 sensors for detecting CO and volatile organic compounds (VOC) were tested and calibrated against an air quality monitoring station in Brazil [38]. The study found that the two sensors added another option into the existing conventional air quality equipment. But, cross-sensitivity problems due to the presence of other gases could occur with these sensors. The authors suggested a multi-sensor system to deal with the cross-sensitivity problem. In another part of the world, four MiCS-5525 sensors were evaluated. Two mathematical models were derived as a result of running two calibrations [39]. The first calibration was conducted in a controlled environment, the second one run in a field test against reference instruments for 19 days, in April 2013. The first calibration had unsuccessfully fitted the output of the reference instruments, while the second one was better in generating optimal model parameter values. Using a non-linear model fitting algorithm, the second calibration incorporated well if heater drift was known. Heater drift is the difference between ambient temperature and surface temperature.

Some research explores the application of low-cost sensor networks based on the mobility of the sensors in the network. High-density sensor networks were explored within stationary-based networks. The idea was to be able to adjust the output and detect an anomaly in the nodes. Tsujita *et al* [40] showcased work on this type of network. The NO_2 sensor networks and an NO_2 instrument analyser were co-located in their study in Japan.

Meanwhile, a smartphone was used as a power source through a USB port with a MiCS-OZ-47 sensor in a non-stationary system, which sensed ozone concentration in the atmosphere [41]. A handheld unit with a few sensors was also proposed to measure air quality in the Common Sense project [42]. The handheld is expected to connect to a smartphone via Bluetooth. The project aims to embed environmental sensors onto a smartphone [43]. Another example of this is the N-SMARTS project [44] which uses commercial off-the-shelf (COTS) sensors deployed easily to any smartphone. Jiang *et al* [45] developed Mobile Air Quality Sensing (MAQS) system and focused their study on indoor air quality by building a sensor prototype working closely with a smartphone.

2.3 Artificial Neural Network

An artificial neural network (ANN) has been applied to forecast where data is made available by many researchers [46]. Speech recognition, image recognition, chemical research, ecological, and environmental sciences are among the numerous applications of ANN mentioned by Lek and Guegan [47]. Gardner and Dorling [48] explained the use of the algorithm in the atmospheric sciences. The growing number of ANN applications has encouraged us to include the algorithm in this thesis.

ANN algorithms model the biological neurons of a human brain processing information. Each neuron is connected to other neurons; therefore, a neuron could receive signals from other neurons, process them, then forward the information to other neurons. The biological network of neurons is random, or, perhaps shows a pattern, but is not easily detectable. An ANN algorithm imitates the concept, referring to a neuron as a node. The connections between nodes are called edges, they have a direction and a weight. Based on

the architecture, the ANN algorithm can be categorized by either feed-forward or recurrent (feedback) networks [49]. Single-layer perceptron, multilayer perceptron, and radial basis function nets are examples of feed-forward architecture. Meanwhile, competitive networks, Kohonen's SOM, the Hopfield network, and ART models are examples of recurrent nets. Feed-forward networks have no loops, while recurrent nets do have loops (feedback mechanisms). The ANN network is not as complex as the human brain, but works by inferring a general model from the data. The algorithm has a number of inputs (or features) and its associated constants (also known as weights) to be fitted into one or more outputs within the training and testing phases. The challenge of ANN is to find an appropriate network layout by determining appropriate weights with optimal fitting performance. This process runs in the training phase. When a right network configuration has been determined, the model would then be used to predict the output based on a number of inputs in the testing phase. The ANN works best with a high number of data and features.

ANN algorithms have been used and deployed widely to many fields for more than a decade. In terms of environmental air quality, for example, Gardner and Dorling [50] used an ANN to predict NO_x and NO_2 concentrations on London's busy roads using monitoring sites in Central London. Nagendra and Khare [51] predicted NO distribution from vehicles in Delhi. The algorithm has also been used in generating a model to calibrate low-cost sensors economically. The trained ANN models were used to adjust the output of tin dioxide (SnO_2) sensors for sensing methane and carbon monoxide [52], and ozone and nitrogen dioxide [53]. ANN models were also used in a detection of VOC [54]. Another interesting study exhibited an alternative ANN technique, selectively picking 5 out of 30 features and analysing data sensors from EOS^{835}. The study argued that optimal performance can be achieved by determining only important features or feature selection [55].

2.4 Anomaly Detection

The quantification of gaseous concentration by low-cost sensors can possibly be enhanced by using redundant sensors. A dense deployment of low-cost sensor nodes creates the potential for every node to be an auditor for their adjacent nodes, and give detailed coverage of the condition of whole areas easily. As environmental awareness is increasing, low-cost environmental sensors will soon be found everywhere.

The data collection process for wireless sensor networks may well include abnormal observation in the sets of data. Abnormal observations may result from noise and errors (i.e. mechanical faults, instrument error), actual events (i.e. changes in system behaviour), human error, or malicious attacks. Atypical observations can potentially affect the data set and the analysis process. Therefore, it is important to set apart these outliers from the population. Although there are different definitions of an outlier, two classic definitions seem to be widely accepted [56]. In statistics, an outlier in an observation can be defined as data inconsistent from the rest of population [57]. Another definition of outlier based on Hawkin's paper [58] is any output with a different mechanism, from many observations. Neyman and Scott [59] argued that certain data distributions might generate outliers. Starting from Neyman and Scott's study, Green [60] further developed six models of statistical distributions based on outlier properties found on the right tail of the distributions. The models classified the distributions to being either outlier-prone or outlier-resistant.

Outlier detection has been studied in many fields, including statistics, data mining, and machine learning. Hodge and Austin [16] provided comprehensive outlier detection methods in their paper. Outlier detection is also known as anomaly detection, deviation detection, novelty detection, or exception mining in the literatures. Hodge and Austin categorized the

characteristics of outlier detection methods based on the use of: pre-labelled data, parameters of data distribution, and type of data set. Evaluation methods can be used as outlier detection method according to the two authors. Given “normal” and “abnormal” as a pre-defined label to the data, outlier detection methods can be distinguished as unsupervised, supervised, and semi-supervised. Statistically, outlier detection can be further divided based on the use of parameters of data distribution: parametric, non-parametric, and semi-parametric methods [61]. On the other hand, multi-layer perceptron, self-organising maps, radial basis function networks, support vector machines, Hopfield networks, and oscillatory networks are among the neural network methods implemented as outlier detection techniques [62].

We are interested in exploring the Autoregressive Integrated Moving Average (ARIMA) model to detect outliers. ARIMA has been known widely in forecasting and it is a well-known concept, used by Chang *et al* [63] to detect outliers. The performance of ARIMA is less than ANN in predicting daily air quality concentration, as pointed out by Prybutok *et al* [64]. However, we look at another possibility to employ the ARIMA model by using it for low cost sensors in a spatial-temporal setting with redundant nodes. The next chapter describes our experiment, using the ARIMA model.

Chapter 3: Experiment

There are three questions to be answered from this experiment:

Q1. How much do readings vary between different AQE sensor products?

Q2. By adding redundant sensors, can the outlier module identify outliers?

Q3. Could we find a mathematical model for AQE in the adjustment module that can fit a reference device?

The experiment was run in three stages: data gathering, data analysis, and data testing. The data gathering phase monitored the environment for a period using the AQE. This data was then analysed, evaluated, and compared to that obtained with the instruments owned by Environment Canterbury, Christchurch, New Zealand. The process of data analysis involves the learning of our proposed methods from the outlier and adjustment modules. The trained methods are then evaluated in the testing phase.

This chapter starts with the description of AQEs, followed by how we design the experiment, obtain and treat data.

3.1 Air Quality Eggs

The AQEs we used are a product of a UK-based start-up company called Wicked Device. They use the MQTT protocol and rely on a ‘wireless fidelity’ (Wi-Fi) connection to publish their readings. The scenario and technical specification are explained further in the following sub-section.

3.1.1 Data Gathering

An AQE publishes the measurements of its surroundings to an MQTT broker on http://mqtt.opensensors.io using a publish/subscribe approach [65]. To publish and subscribe AQE data to OpenSensors, an AQE device requires four properties to be set, which are: an API-key (determined by OpenSensors), a device *client-id*, a device password (determined by OpenSensors), and a unique *topic id* which has the pattern of:

/users/<Username>/<Topic_Name>.

There are two options available for data publishing and subscribing: the mosquitto client or the HTTP protocol. AQE installs a program call mosquitto client on its Arduino board. The mosquitto client sends mosquitto_pub and mosquitto_sub commands to publish and subscribe to the MQTT broker on the following URL: https://mqtt.opensensors.io/ws. Both commands require passing the same number of parameters. The syntax to publish data, or subscribe data differ only in the command, using the mosquitto client in the following:

Mosquitto_pub –h mqtt.opensensors.io –i <DeviceID> -t /users/<UserName>/<TopicName> -m <Message> -u <UserName> -P <Device Password>

Where:

- -h specifies the OpenSensors broker. In this example: mqtt.opensensors.io
- -i specifies device id
- -t specifies topic name. A conventional pattern is /users/<UserName>/<topic>
- -u specifies user name
- -P specifies device's password
- -m specifies your published message

The developers can use Paho Javascript and MQTT over WebSockets to replace the use of the mosquitto client. The developers can also programmatically use Javascript, Python, or curl to connect to the MQTT broker via HTTP protocol.

3.1.2 Technical Specification

An AQE has three sensors, namely the MICS 2710 (NO_2 detector), MICS-5525 (CO detector), and AM2303 (sensing temperature and humidity) [66]. The MICS series sensors are semiconductor sensors [67], while the DHT22 is a capacitive-type sensor [68].

The MICS sensors have a sensing layer that works by combining the thermal, chemical and electrical effects. To begin a sensing operation, the sensing layer has to be heated first, known as the warm-up phase. The gas will affect the electrical resistance in the sensors' sensing layer, which can then be compared with the stored reference value. Due to their characteristics, several problems are present in the application of MICS sensors. First, the minimal resistance level may differ between each sensor. Second, other gases are likely to interfere with the targeted gas. Third, frequent use of the sensor may affect its sensing layer. Fourth, an appropriate temperature is required for the sensor to measure the gas accurately. The last problem is that the ambient temperature and humidity may affect the baseline, the sensitivity, and reactivity of the sensors. Therefore, the use of these sensors requires extensive calibration.

Although it has the same function of sensing temperature, the DHT22 sensor is different from the AM2303 sensor in the output interface, which uses pins instead of wires [69]. The sensing element of the DHT22 sensor comprises a polymer humidity capacitor and DS18B20 (detecting temperature). It connects to an 8-bit single-chip computer. Aosong Electronics, the vendor of DHT22 sensors, claims that they are calibrated to obtain a calibration-coefficient [68], which can be used to compare with the value of the sensor readings. The application of DHT22 sensors needs to consider operating and storage conditions; vapour from chemical materials, for example, could potentially affect the

sensitivity of the sensor. A change of temperature, particularly as a result from other parts in the device, may affect the measurement. Lastly, strong light and ultraviolet may degrade the performance of the sensor.

The sensors specifications are available online. Wicked Device claims they calibrate the sensors before sending the products to end users.

3.2 Apparatus

The three AQEs used to monitor air quality in this study are brand new. Figure 3.1 shows the installation of AQEs on site during the experiment from 12 September 2016 09:39 AM until 4 December 2016 11:59 PM

Figure 3.1 The placement of three Air Quality Eggs on top of a roof at the Riccarton Road site.

3.3 Deployment

The three AQE devices were attached to a pole and placed in a line on top of a white power box. The box was located on the top of the roof in the Environment Canterbury's Riccarton Road site. The AQEs connected to a 3G modem inside the box, provided by Vodafone. Power was supplied from the box to the modem and the three AQEs via a USB port. A weather-proof tape was used to seal any joints in the AQEs, preventing any water

seeping in. The AQEs ran without any supervision or maintenance during the experiment. Vodafone's 3G modem acted as an access point because the AQEs require a Wi-Fi connection for publishing the measurement values. The download and upload speeds of the modem on the site were 7.86 Mbps and 1.86 Mbps, respectively. The power box leaned on the wall to maintain its position against strong winds, despite its heavy mass. The shade of a 2-level building next to the site shielded the AQEs from direct sunlight during daytime.

The AQEs were placed on the same site as the sophisticated instrument analysis operated by Environment Canterbury (ECan) at the Riccarton Road site in Christchurch, New Zealand. The CO reference meter is a Gas Filter Correlation CO Analyzer Model 300E, and the NO_2 reference meter is Chemiluminescence NOX Analyzer Model 200E. Both analysers were calibrated with the corresponding gas every month. The AQEs record the air every 5 seconds, while the Reference Meter does so every hour. The AQE results are averaged into hourly readings in order to be compared with the result of the Reference Meter.

Both ECan and AQE data obtained from the data collection phase were trained and tested on two different computers, using R, specifically using RStudio. Both machines have the following specification: 64-bit 8GB RAM and i7 processor. The two machines' operating system and R software are: MacOS 10.11.6 running RStudio version 0.99.903 and Linux Mint Cinnamon 2.8.8 running RStudio version 0.99.491.

3.4 Data Sources and Treatment

We monitored the air quality at Riccarton Road using AQEs in the data collection. The supervised learning method was applied to the data training and data testing phases for the proposed outlier and adjustment modules. The dataset obtained from the first stage was split

into two parts for data training and data testing purposes. The first part of the dataset was utilised to generate models in the data training phase. The generated models were then evaluated in the data testing stage using the second part data.

The ECan data was used as the main reference. Two different ECan datasets were used to be analysed and evaluated: past and current ECan. The past ECan data contains hourly measurements from the sophisticated instrument analysis on the same site, between 2014 and 2015. The data is used to train the outlier module for generating ARIMA models. The current ECan data is used to evaluate the outlier in the testing phase, and the adjustment modules in both the training and testing phases. The current ECan data was obtained from the same period of the AQEs and was averaged hourly.

Both past and current ECan data are accessible through the Environment Canterbury website [70]. The past ECan data is limited to only the last two years (2014-2015). The ECan data has 5 columns: DateTime, StationName, Temperature, CO (mg/m3), and Relative humidity (%). Past ECan data must first be converted into a suitable format accepted by R. There were three treatments to the ECan data. Firstly, the DateTime field is not in an R standard time format since ECan data uses the a.m./p.m. format. The accepted format in R is AM/PM. Secondly, the StationName only indicates Riccarton Road. Thus, this column was eliminated. Finally, the date field was not in a sort order. Hence, the data was sorted by date.

WickedDevice, the producer of AQE, restricted the data to downloads every 7 days. Downloaded files contained weekly periods and divided to the following weekly and monthly batches:

- The first week: 12 September 2016 00:00 to 18 September 2016 23:59
- The second week: 19 September 2016 00:00 to 25 September 2016 23:59

- The third week: 26 September 2016 00:00 to 2 October 2016 23:59
- The fourth week: 3 October 2016 00:00 to 9 October 2016 23:59
- The fifth week: 10 October 2016 00:00 to 16 October 2016 23:59
- The sixth week: 17 October 2016 00:00 to 23 October 2016 23:59
- The seventh week: 24 October 2016 00:00 to 30 October 2016 23:59
- The eighth week: 31 October 2016 00:00 to 6 November 2016 23:59
- The ninth week: 7 November 2016 00:00 to 13 November 2016 23:59
- The tenth week: 14 November 2016 00:00 to 20 November 2016 23:59
- The eleventh week: 21 November 2016 00:00 to 26 November 2016 23:59
- the twelfth week: 26 November 2016 00:00 to 4 December 2016 23:59 AM.
- The first month: 12 September 2016 00:00 AM to 10 October 2016 00:00 AM
- The second month: 10 October 2016 00:00 AM to 7 November 2016 00:00 AM
- The third month: 7 November 2016 00:00 to 5 December 2016 00:00 AM

Due to an intermittent internet connection on the site, AQE data is missing from the following periods:

- 25 September 2016 23:00
- 11 October 2016 01:00 until 14:00
- 26 November 2016 16:00 until 29 November 2016 14:00
- 30 November 2016 03:00 until 1 December 2016 15:00

AQEs with identities egg008027a2bc1b0113, egg008027a278880113, and egg008028735b980112, were defined, respectively, as AQE1, AQE2, and AQE3. These identifiers are referenced in the tables and figures.

Of all weekly downloaded AQE data files, a change was imposed in a name of a column, particularly to the last column titled “altitude[m]”. It seems that there was a change in naming the column on the third month of the experiment. The new column was “altitude[deg]”. Although the column was unused, the renaming affects the code in R. Therefore, the old column name was used to refer to the last column in the AQE data.

The AQE dataset has 10 columns, of which half are relevant: timestamp (in mm/dd/yyy hh:mm:ss format), temperature (in degrees Celsius), humidity (in percentage), NO_2 (in ppb), and CO (in ppm). The other five columns: NO_2 (in volt), CO (in volt), latitude (in degree), longitude (in degree), altitude (in meter) are ignored. The total number of 5-seconds readings on AQE1, AQE2, and AQE3 are 1,253,363, 1,255,578, and 1,256,345 rows, respectively. Averaging the 5-second reading into one hour, there were 2,006 hours. Due to an intermittent internet connection, 8% of hourly data was missing but it did not affect the result as we omit the missing data in the calculation. The outlier module is explained in the next chapter.

Chapter 4: Outlier Module

We are interested in exploring statistical models in the outlier module based on Hodge and Austin's classification [16]. The ARIMA method is used and the calculation of the ARIMA is then compared to the neighbour sensors' data to classify whether a value is an outlier or not. Supervised learning is used to generate the appropriate ARIMA model to be employed by the four proposed decision schemes in the outlier module. This chapter starts with introducing the four proposed decision schemes for detecting an outlier and the evaluation methods in Section 4.1. Then, the process to generate ARIMA models using the Box-Jenkins approach are discussed in Section 4.2. The training of the ARIMA models uses the 2014 and 2015 data from ECan, and the difference between sensors are also discussed in Section 4.3 to Section 4.5. Finally, the research findings and the evaluations of the decision schemes using AQE readings and ECan data are discussed on testing phase in Section 4.6.

4.1 Proposed Decision Schemes

Four decision schemes are proposed and examined in the outlier module. They basically can be distinguished based on whether they check the neighbour sensor or not. The input to each scheme is the AQE dataset, which is divided into weekly or monthly batches, described in Section 3.4. A scheme predicts the values of the next batches based on an ARIMA estimator, which should be on the prediction range of the scheme. The prediction range is a range of the expected value plus or minus one standard deviation.

The four proposed decision schemes for outlier modules are explained below.

1. Static Detection scheme.

The first batch of AQE data is the input to an ARIMA model and the outcome is the forecast for the next two batches ($\hat{y}_i$). Algorithm 4.1 illustrates the algorithm, while the standard deviation and the mean are calculated based on Equation 4.1 and Equation 4.2. The predicted value is then compared to the second and the third batch of AQE data as a reference. When the prediction ($\hat{y}_i$) and AQE (y_i) are compared to each other, the scheme marks an outlier if a reference is not within a range of prediction and its standard deviation. The algorithm for marking the outlier is explained below. Figure 4.1(a) illustrates the Static Detection scheme. Dotted blue lines show the comparison between predicted ARIMA model and AQE data. It also represents the evaluation of the scheme. The evaluation method is explained in section 4.2.

#using AQE's first batch predicts the output of two batches ahead

#compare predicted value ($\hat{y}_i$) and AQE data (y_i) of two batches ahead *Algorithm 4.1.*

For each i^{th} row:

Within range: if $(\hat{y}_i - \sigma) < y_i < (\hat{y}_i + \sigma)$

Mark as outlier: if $y_i > (\hat{y}_i + \sigma)$ OR $y_i < (\hat{y}_i - \sigma)$

Standard deviation (σ) is calculated as:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu)^2} \qquad \textit{Equation 4.1}$$

where:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad \textit{Equation 4.2}$$

2. Dynamic Detection Scheme.

Figure 4.1b renders the Dynamic Detection scheme with dotted blue lines show the values comparison and the evaluation of the scheme. An ARIMA model is dynamically

calculated between the two consecutive batches. The previous batch is used to predict the later one. The calculation goes on until the last batch of the data (the twelfth week in the case of weekly input and the third month for monthly). Contrary to the Static Detection scheme with only one model for comparison to other data batches, the Dynamic Detection scheme is calculated dynamically between two consecutive batches (either monthly or weekly).

Take the first and second monthly batches as an example. The input of an ARIMA model is the first month of AQE data and the output is the prediction for the second month of AQE data ($\hat{y}_i$). Similar to the Static Detection scheme, both prediction and reference are compared to detect outliers in the Dynamic Detection scheme. The same Algorithm 4.1, Equation 4.1, and Equation 4.2 are used. The predicted value (y_i) is then compared to the second batch of AQE data as a reference (y_i). When the prediction ($\hat{y}_i$) and AQE (y_i) are compared to each other, the scheme marks an outlier if a reference is not within a range of prediction and its standard deviation. The scheme proceeds to calculate, predict, and evaluate the next batch until the last batch.

3. Dynamic with Comparison to Neighbour Scheme

This scheme is similar to the Dynamic Detection scheme. However, when the scheme detects AQE data (y_i) outside the prediction range ($\hat{y}i$) and its standard deviation, the scheme checks a suspected outlier against its adjacent nodes (z_{in}) and their standard deviation. The scheme marks a value as an outlier when the value (y_i) is still out of the range on all adjacent node values (z_{in}) and their standard deviations.

Algorithm 4.2 and Figure 4.1c illustrate the proposed method.

#using AQE's first sequence predicts the output of one batch ahead

#compare predicted value ($\hat{y}_i$) and AQE data (y_i) of one batch ahead Algorithm 4.2.

For each i^{th} row:

Within range: if $(\hat{y}_i - \sigma) < y_i < (\hat{y}_i + \sigma)$

Possibility of an outlier: if $y_i > (\hat{y}_i + \sigma)$ OR $y_i < (\hat{y}_i - \sigma)$

#Checks values (z_i) of n neighbours (z_{in}) and their standard deviation (σ_n)

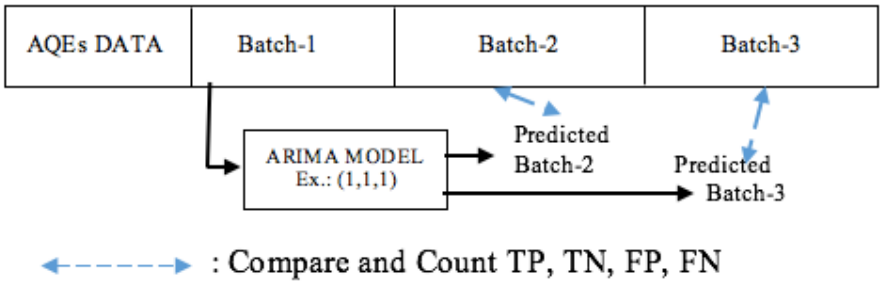
For each n^{th} neighbour:

Possibility of an outlier: if $y_i > (z_i + \sigma_n)$ OR $y_i < (z_i - \sigma_n)$

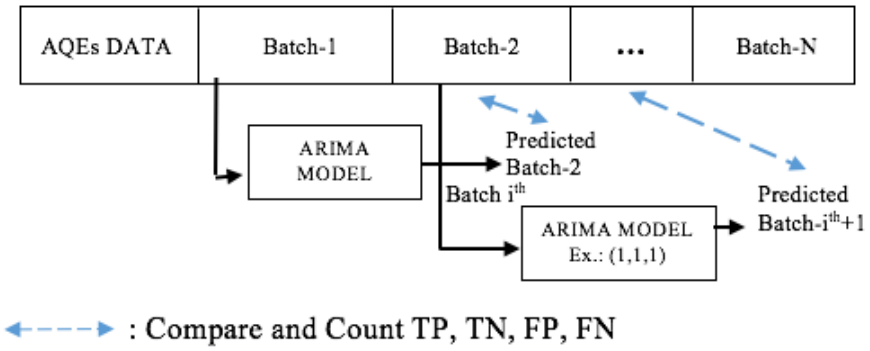
Marks as an outlier if y_i is out of the range for all neighbours

4. *Dynamic with Comparison to Neighbour's ARIMA model Scheme*

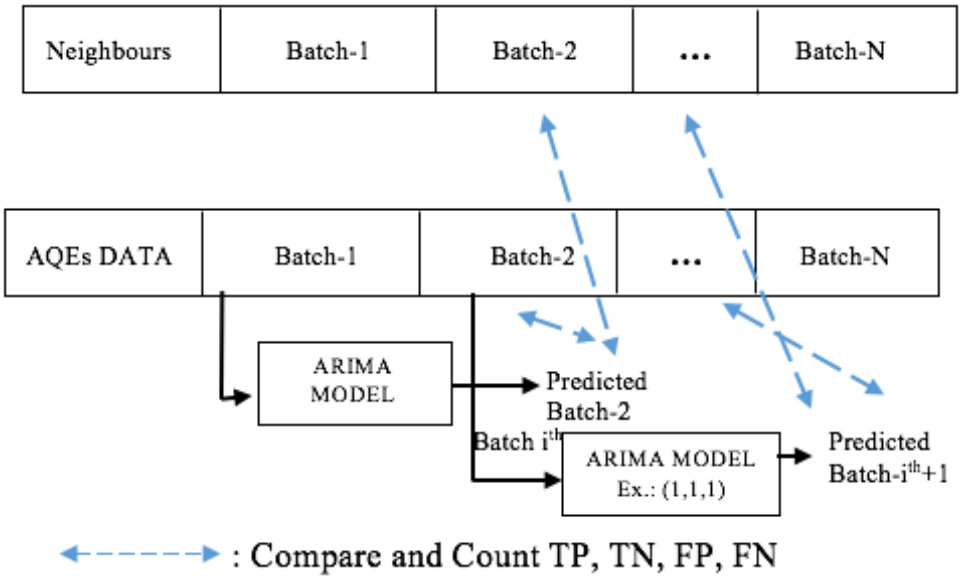
This scheme is similar to the Dynamic with Comparison to Neighbour scheme, but it differs in evaluating their neighbours. The scheme calculates their ARIMA model and compares the predicted values against suspected outliers when evaluating the neighbours. The scheme marks the outlier only if the suspected outliers are out of the range on all predicted neighbours' ARIMA model. Figure 4.1d illustrates the scheme.



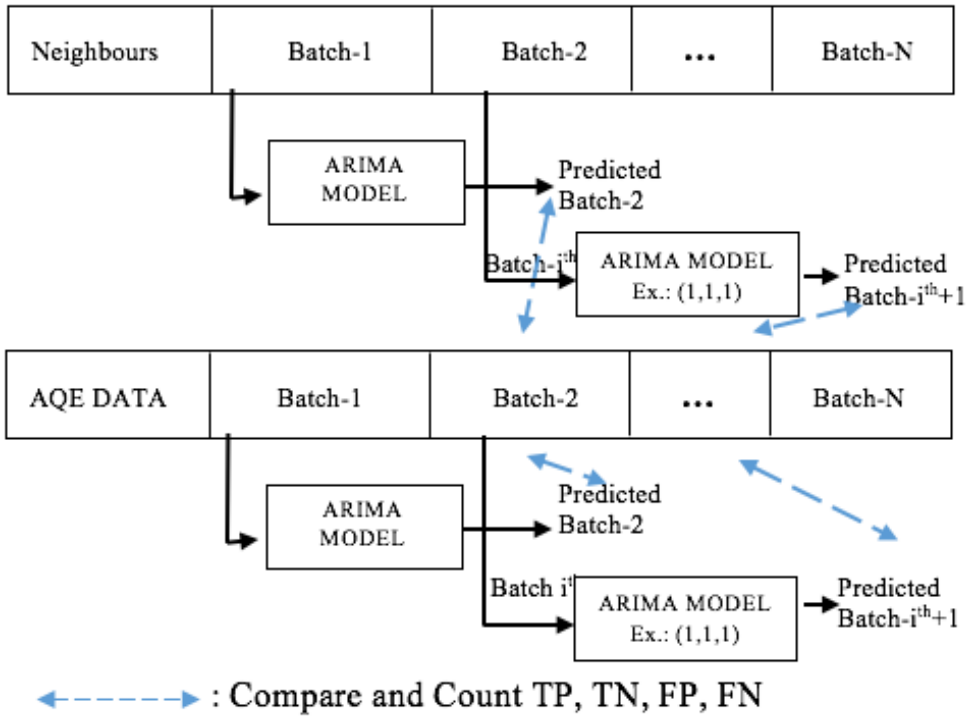
(a)



(b)



(c)



(d)

Figure 4.1 Proposed decision schemes in the outlier module (a) static detection. (b) dynamic detection. (c) dynamic detection with comparison to neighbours. (d) dynamic detection with comparison to ARIMA's neighbours

4.2 Evaluation Methods

Classifying whether a reading is an outlier or not is a challenging task, particularly because a sudden change in the air quality dataset can be correlated to a sudden change in environment, instead of the sensor's fault. We apply the Type 3 approach to the problem of outlier detection [16]. Type 3 assumes only normal data with few outliers exist in the dataset. Outliers are then assessed statistically using the proposed detection schemes in order to find them in the data, particularly when a value is outside the boundary. The proposed detection schemes identify values outside their prediction range as outliers.

Classification accuracy is used in assessing the performance of each decision scheme. Classification accuracy is assessed by counting the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The evaluation method requires a reference to identify whether the schemes have correctly classified the values. TP indicates that a scheme has correctly identified a value as normal data, compared to the reference. TN indicates the scheme has correctly identified a value as an outlier, FP means the scheme has incorrectly identified the value as normal data, while FN indicates the value has incorrectly identified as an outlier. The assessment of a scheme is calculated as:

$$Classification\ Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad Equation\ 4.3$$

4.3 ARIMA Models

Time series analysis is used for detecting any outliers found in the AQE's output. A time series is an ordered series observed in a certain period [71]. The time domain analysis assumes that there exists a dependent correlation between the current value and past values [72]. The current and past values act as inputs to a model for predicting future values of a

time series. Regarding the time series analysis, Shumway and Stoffer [72] discuss three possible approaches: linear regression, Box-Jenkins, and additive models. They also mention other methods: frequency domain, and a combination of time series and frequency domain approaches. The use of time series can be found in many applications, such as engineering, agriculture, meteorology, and quality control.

A univariate outlier detection model is implemented by employing the Box-Jenkins approach. The method is also known as the ARIMA (Auto Regressive Integrated Moving Average) model and is discussed in detail below.

4.3.1 Time Series Model with Seasonal ARIMA

Stochastic models can be used to calculate the probability of a future value in the range of two specified limits [73]. One class of stochastic models is stationary models. Stationary models require fixed probabilistic processes, meaning that the distribution function of the stationary series does not change over time. Stationary models can be distinguished based on the mean and variance. A model can be categorized as a strictly stationary process when its mean is zero. But, strictly stationary processes never exist in the real world. A model with a fixed constant mean and constant variance is categorized as a weakly stationary process. Nonstationary models, as opposed to stationary models, tend to fluctuate in amplitude, so they may not have a constant mean level over time. Nonstationary models are the most common models found in many time series problems.

A general ARIMA model combines the past values, the difference between past values, and forecast error to predict the future values in the time series problems. ‘Auto regressive’ takes into consideration past values which contribute to the future values. ‘Moving average’

predicts future values by calculating its forecast errors and averaging out noise. ARIMA models can only be applied to stationary processes. When the series is a non-stationary process, it requires to be transformed first into a stationary process by applying the difference to the series. This process is said to be “Integrated” in the ARIMA model.

Ideally, the model of autoregressive (AR) and moving average (MA) representations contains an infinite number of available observations. However, this notion is not applicable in the real world due to data limitation. The model for the time series has a finite number of parameters. An autoregressive process of order **p** means that the AR model is limited to a number of *p* observations in the past, while a moving average process of order **q** can be interpreted as *q* number of lag forecast errors from the past observations. **d** notation represents the number of differencing steps.

Seasonal ARIMA is introduced when the given series shows a repeated or seasonal pattern. Suppose that five-year precipitation data shows heavy rain every Spring, drizzle in Summer, or a dry Autumn. This phenomenon can be explained using a seasonal ARIMA model. Capital letters are used to denote the order of the seasonal model. **P** means the number of past values for seasonal AR, **Q** is the number of the lag forecast error, and **D** indicates a number of differencing steps imposed to the seasonal time series. Describing the seasonal ARIMA model, the notation of the model can be written as (p,d,q) x (P,D,Q).

Equation 4.4 is a seasonal ARIMA model with four types of constants (θ,ϕ,Θ,Φ) where $\phi_p(B)$ and $\theta_q\,(B)$ are the regular autoregressive and moving average factors and $\Phi_P\,(B^s)$ and $\Theta_Q\,(B^s)$ are the seasonal autoregressive and moving average factors, respectively [71]. $(1\text{-}B)$ indicates a one-time difference. Equation 4.4 can be written as ARIMA (p, d, q) x (P, D, Q)$_s$ where sub-index *s* refers to the seasonal period.

$$\Phi_P(B^S)\phi_p(B)(1-B)^d(1-B^S)^D\dot{Z}_t = \theta_q(B)\Theta_Q(B^S)a_t \qquad \textit{Equation 4.4}$$

where:

$$\dot{Z}_t = \begin{cases} Z_t - \mu, if\ d = D = 0 \\ Z_t, \qquad otherwise \end{cases}$$

$$a_t = forecast\ errors\ from\ the\ past\ observations\ at\ time\ t$$

$$\phi_p(B) = autoregressive\ factors = 1 - \phi_1 B - \phi_2 B^2 - \phi_p B^p$$

$$\theta_q(B) = moving\ average\ factors = 1 - \theta_1 B - \theta_2 B^2 - \theta_q B^q$$

$$\Phi_P(B^S) = seasonal\ autoregressive\ factors = 1 - \Phi_1 B^S - \Phi_2 B^{2S} - \Phi_P B^{PS}$$

$$\Theta_Q(B^S) = seasonal\ moving\ average\ factors = 1 - \Theta_1 B^S - \Theta_2 B^{2S} - \Theta_Q B^{QS}$$

B is a backshift operator $B^j x_t = x_{t-j}$. It is used to shorten the equation. Equation 4.4 has been shortened because of the **B** operator. Suppose that we use the temperature model of (2, 2, 2) x $(1, 1, 1)_{12}$ or (2, 2, 2) x (12, 12, 12). Each element from Equation 4.4 for the given ARIMA model can be written as:

$$\phi_2(B)Z_t = 1 - \phi_1 Z_{t-1} - \phi_2 Z_{t-2}$$

$$\theta_2(B)a_t = 1 - \theta_1 a_{t-1} - \theta_2 a_{t-2}$$

$$\Phi_2(B^{12})Z_t = 1 - \Phi_1 Z_{t-12}$$

$$\Theta_2(B^{12})a_t = 1 - \Theta_1 a_{t-12}$$

$$(1-B)^2 Z_t = Z_t - Z_{t-1} - Z_{t-2}$$

$$(1-B^{12})^1 Z_t = Z_t - Z_{t-12}$$

From the seasonal ARIMA model above, Z_t is current temperature at time *t*, Z_{t-1} is the past temperature value before Z_t, Z_{t-2} is the past temperature before Z_{t-1}, and Z_{t-n} is the past output of *n* observation before Z_t.

4.3.2 Box-Jenkins Approach

Time series analysis is a tool to capture the nature of past processes in order to predict future values by the use of mathematical model and data analysis. To generate a model, a theoretical and data analysis is usually combined to perfect the model. The Box-Jenkins approach is used as a starting point. Several ARIMA models may need to be evaluated and explored to choose the best fitted model, identified in three steps: model identification, parameter estimation, and diagnostic checking.

A. Model Identification

Four stages are involved for model identification [71]. The first step is plotting the time series data in order to determine whether the series is a stationary or non-stationary process. Data differencing is applied when it is a non-stationary process. The next three steps are: calculate the auto correlation function (ACF), calculate partial auto correlation function (PACF), and test the deterministic trend term θ when *d > 0*.

Both ACF and PACF are useful in determining the order of differencing, autoregressive, and moving average polynomials. ACF is used to determine the appropriate number of lagged error terms, while PACF identifies the order of an autoregressive model. The ACF denotes covariance and correlations of a point at time *t* (z_t) and itself at different points (z_{t-k}, for k=1, 2, 3, etc). The PACF shows the correlation between two points in the series, z_t and z_{t+k}, given that all other points or observations between these two points ($z_{t+1}, z_{t+2}, \dots, z_{t+k-1}$) are omitted. The *stats::acf* and *stats::pacf* from the R functions are called in estimating the ACF and PACF. The functions implement the work of Ripley and Venables [74].

B. Parameter Estimation

It is common that some models may have a similar result in the parameter estimation stage [71]. Therefore, some criteria are available to select a model in generating a time series model, including the Method of Moments, Maximum Likelihood Method, Non Linear Estimation, and Ordinary Least Squares Estimation [71]. Dent and Min [76], and Ansley and Newbold[75, 76] have conducted simulation studies to assess the performance of the Conditional Least Squares, Unconditional Least Squares, and Maximum Likelihood techniques for estimating parameters in ARMA models. If an estimator has to apply only one technique, Dent and Min propose Maximum Likelihood over the others. However, the two studies suggest that both Least Squares methods suit larger sample sizes, while Maximum Likelihood works better with small or moderate sample sizes. The result of two simulation studies is strengthened by the study of Hillmer and Tiao [7[77, 78], and Osborn [77, 78]. Later, we discuss the experiment challenge in the testing phase with regards to selecting an appropriate method for parameter estimation in Section 4.6.

R has a built-in function (*stats::arima*) to build the ARIMA model. The Maximum Likelihood (ML) method is chosen to examine the parameters in the training phase. The Exact Likelihood function is used among the other two ML functions: (i) Conditional Maximum Likelihood Estimation, and (ii) Unconditional Maximum Likelihood Estimation and Backcasting Method [71]. The algorithm of Gardner *et al* [79, 80] and the study of Dublin and Koopman [79, 80] are applied to the Exact Likelihood function.

C. Diagnostic Checking

Diagnostic checking tests model candidates and chooses an appropriate one. The methods for diagnostic checking are Akaike's Information Criteria (AIC), Bayesian Information

Criteria (BIC), Schwartz's Bayesian Criterion (SBC), Parzen's Criterion for Autoregressive Transfer (CAT). Akaike Information Criterion is a common method for diagnosing a model and has been employed in many time series programs [71]. We employ AIC in determining the best ARIMA model.

The AIC determines the goodness of a model based on the mean expected log likelihood. It provides a balance between the virtue of fitting the model and the complexity of the model. Sakamoto *et al* [81] implied that a model with the lowest AIC can be chosen. When there are several models with almost the same values, they suggest choosing the model with the smallest number of parameters. The method estimates the information lost in generating the data for a given model. Sakamoto *et al* argued that the differences of AIC values generated from the models under investigation are more important than the actual AIC values themselves. AIC cannot measure the model in the real application and it cannot tell if the models fit poorly. AIC is defined as follows [81]:

$$AIC = -2 \times (\text{maximum log likelihood of the model}) + 2 \times (\text{number of free parameters of the model}) \quad \text{Equation 4.5}$$

4.4 Training Phase

Past ECan data is assessed statistically in obtaining appropriate ARIMA models to be used in the detection schemes. The data is separated based on the types of sensors. The following four subsections discuss the finding of the past ECan data for the training phase.

4.4.1 Temperature

Plotting the two-year temperature between 2014 and 2015, we obtain Figure 4.2. Standard deviation is 5.43 degrees Celsius and the average two years' temperature is 12.29

degrees Celsius, where the minimum values are 0 degrees Celsius and the maximum value is 33.26 degrees Celsius. Looking at each year alone, the mean temperature of 2014 is 12.49 degrees Celsius where the minimum value is 0 degrees Celsius and the maximum value is 32.46 degrees Celsius. Meanwhile, the 2015 mean temperature is 12.63 degrees Celsius with 0 degrees Celsius and 33.26 degrees Celsius as the minimum and maximum temperature, respectively. Note that there are 107 hours of missing data of which 36 hours occurred in 2014.

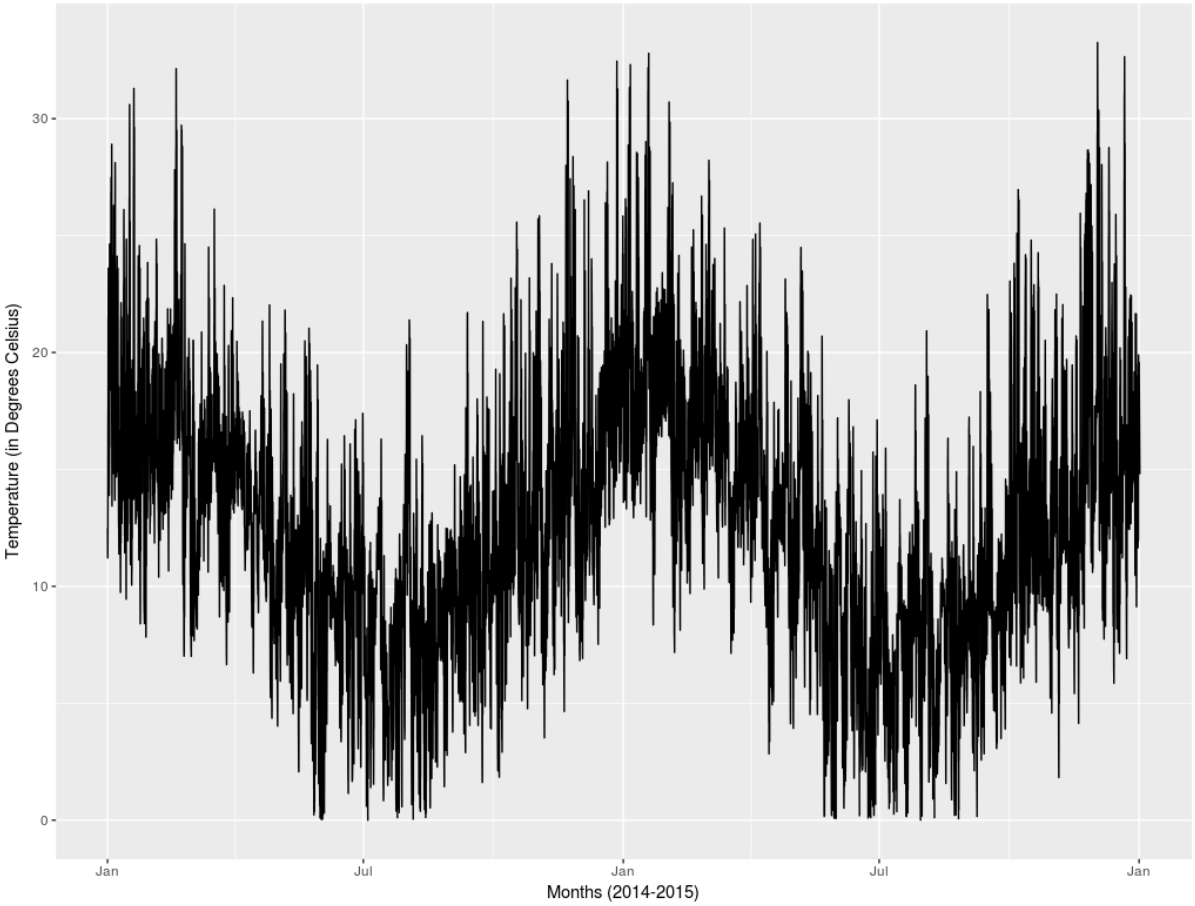


Figure 4.2. 2014 and 2015 temperature plot

Two trends can be observed from Figure 4.2. Firstly, although the annual temperature fluctuated, it returns to some points in the middle. This indicates a weakly stationary process and a seasonal trend. The temperature tends to be relatively warmer in the Spring

(September to November) and Summer (December to February) than in the Autumn (March to May) and Winter (June to August) seasons [82]. We would expect annual temperature series to be symmetrical over long periods. ACF and PACF are computed at the first stage to determine possible models. Figure 4.3 depicts the result of the computation of ACF and PACF where it shows a sustained large ACF with slow decay values, and a large PACF value in the first lag. Although the two-year measurement of temperature indicates a weakly stationary process, Figure 4.3 suggests the temperature needs to be differenced. Data differencing is needed if ACF decays very slowly and PACF cuts off after lag 1 [71]. Therefore, data differencing is applied to the dataset and its result is described in Figure 4.4.

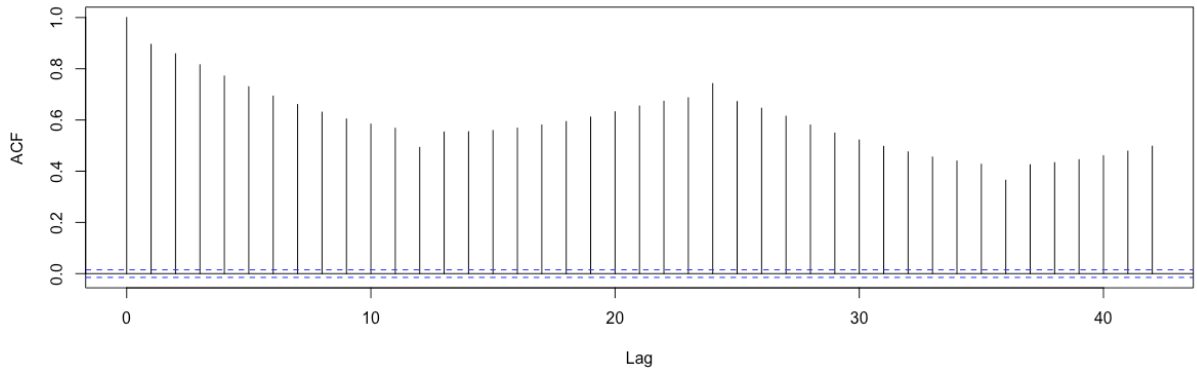


Figure 4.3. ACF and PACF of two years temperature (2014 and 2015)

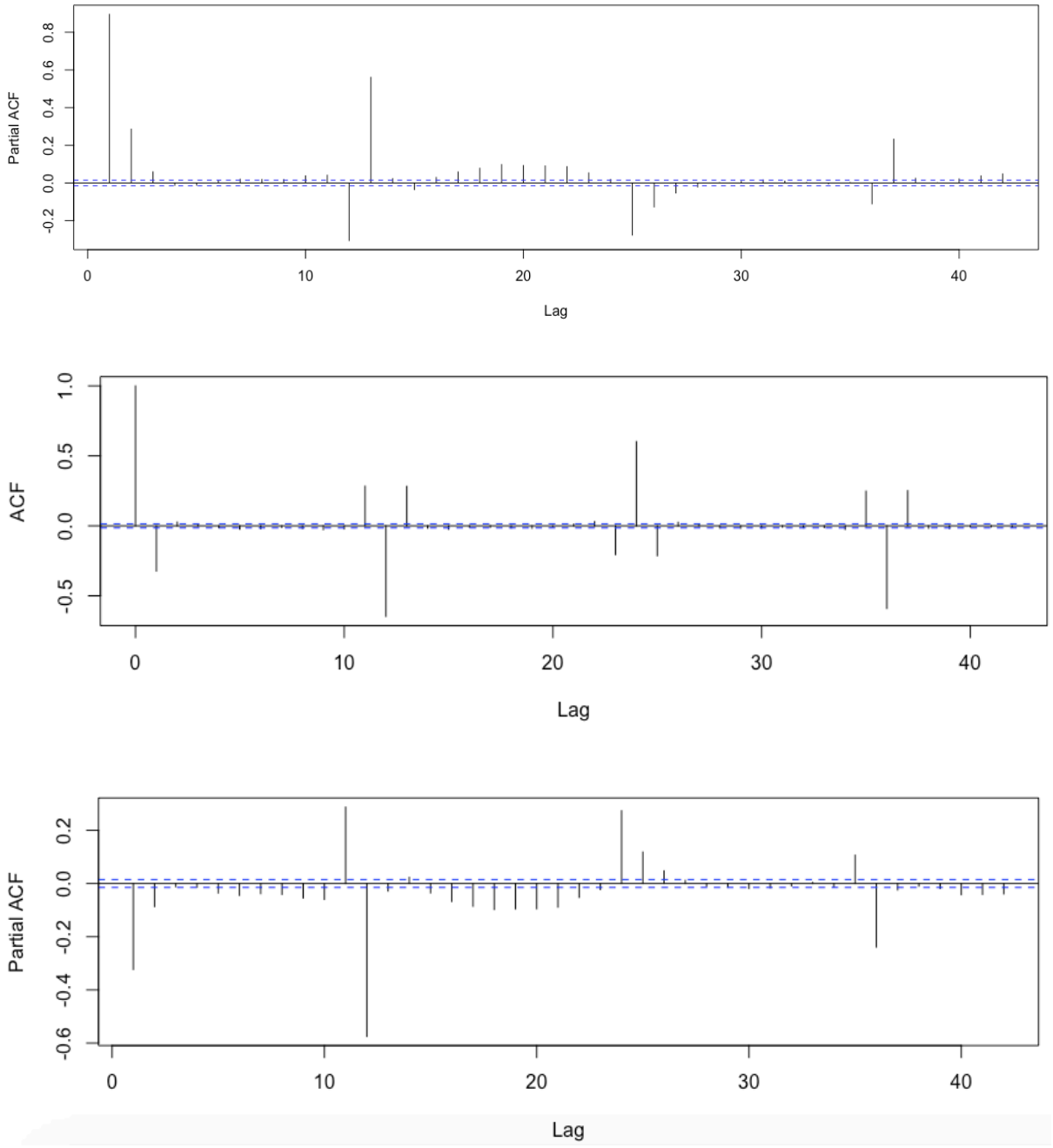


Figure 4.4. ACF and PACF of two years' temperature (2014 and 2015) after one-time differencing

The series in Figure 4.4 is now a stationary process. The ACF and PACF of temperature from Figure 4.4 indicates there is no correlation in the series. However, Figure 4.4 shows first lag and lag 13th has a negative correlation factor of 0.335 and 0.28, respectively. One order differencing of the series seems to show the existence of a seasonal trend as indicated by lag 12th and 24th. Both lags have a factor of -0.64 and 0.59. Figure 4.4 suggests a seasonal 12-

hour, 24-hour, or 36-hour pattern in the series. This seasonal trend is confirmed in the PACF plot. Lag 12^{th}, 24^{th}, and 36^{th} correlate to current time with the respective factor of -0.57, 0.27, and -0.24. From the plotting of ACF and PACF in Figure 4.4, zero or first order of AR, first order of MA, and either 12, 24, or 36 order of seasonal ARIMA model will all be candidates for being the model.

Although we already have a good starting point in deciding which ARIMA model is the best, estimating a model can still be a challenge. We might need to examine all possible configurations of the model. The Box-Jenkins approach emphasizes ACF and PACF are useful in model identification, but there may be an occasion where we need to assess all possible orders of autoregressive and moving average. Obtaining earlier indications of p, d, and q values, we examine all possible models and obtain the AIC values. Table 4.1 shows the result of the calculation. Note: A single asterisk means the parameters are near the edge of the stationarity region. The Maximum Likelihood method in the parameter estimation requires that the ARIMA model must be in a stationary process and the single asterisk indicates that the calculation has a potential to be out of the region. A double asterisk indicates that the estimation cannot be continued as the calculation leads to infinity. A triple asterisk depicts that the estimation takes exhaustive time for calculation.

p	d	q	P	D	Q	AIC
1	1	1				79,079
1	1	4				78,191
1	1	4	12	1	12	65,611
1	0	0				81,007
5	1	4				75,780
5	1	5				74,750
2	1	2	2	1	2	78,217 *
2	1	2	2	0	2	79,091 *
1	1	1	1	1	1	80,827 *
5	1	4	1	0	1	75,478
5	1	4	1	1	1	75,370

1	1	1	12	0	12	**
1	1	1	24	0	24	63,161 *
1	1	1	12	1	12	65,687
1	1	1	24	1	24	64,216 ***
0	1	1	24	1	24	**
0	1	1	12	1	12	66,160
1	1	0	12	1	12	67,405
0	1	0	12	1	12	66,315 *

Table 4.1 Diagnostic checking for temperature models using AIC.

The AIC method is used to decide which p, d, q values are best to use as a model. We conclude that ARIMA (1,1,1) x (12,1,12), (1,1,1) x (24,0,24), and (0,1,1) x (12,1,12) have the smallest AIC value. The performance of the three models would be decided later based on the evaluation of the model on the outlier module.

4.4.2 Carbon Monoxide (CO)

The hourly CO level in Riccarton Road over the last two years is depicted in Figure 4.5. The figure shows little variations of CO gas concentration in the daily measurement. The standard deviation is 0.7 ppm and variance is 0.5 ppm. The total average of two-year measurement for CO concentration is 0.7 ppm with 0 ppm being the minimum value and 46.8 ppm as the maximum value. The annual average of CO concentration is 0.7 ppm, where 0 ppm and 6.4 ppm are the minimum and maximum value of hourly CO concentration in 2014. The hourly variation is relatively small by 0.4 ppm. Meanwhile, the 2015 average of hourly CO concentration is 0.5 ppm within the range of 0 ppm and 46.8 ppm. The variance is 0.6 ppm. There seems to be a pattern in the hourly reading of CO concentration. The CO concentration tends to be higher starting in the late of Autumn and ending in the beginning of Spring. There seems to be an anomaly on 15 July 2015 between 3 pm and 4 pm. The concentration of carbon monoxide has suddenly spiked but we cannot explain the event due to lack of data. To

understand the event, we may need to look at the possibility if there was an accident, or heavy traffic on the site.

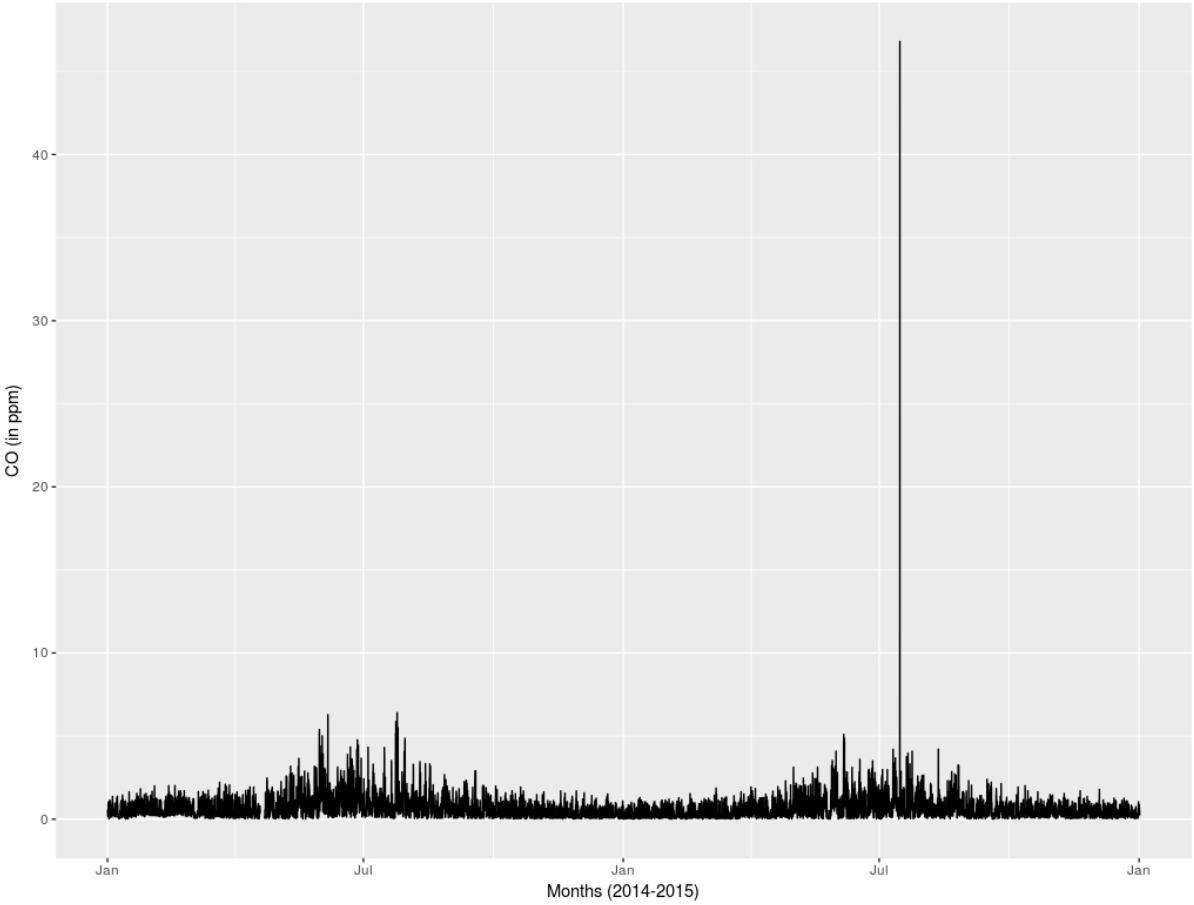


Figure 4.5 2014-2015 readings of hourly CO concentration.

Figure 4.5 indicates the series of CO is weakly stationary with a seasonal trend. The plotting of ACF and PACF for CO process as depicted in Figure 4.6 suggesting a series differencing. Figure 4.7 shows ACF and PACF for CO after one order of differencing.

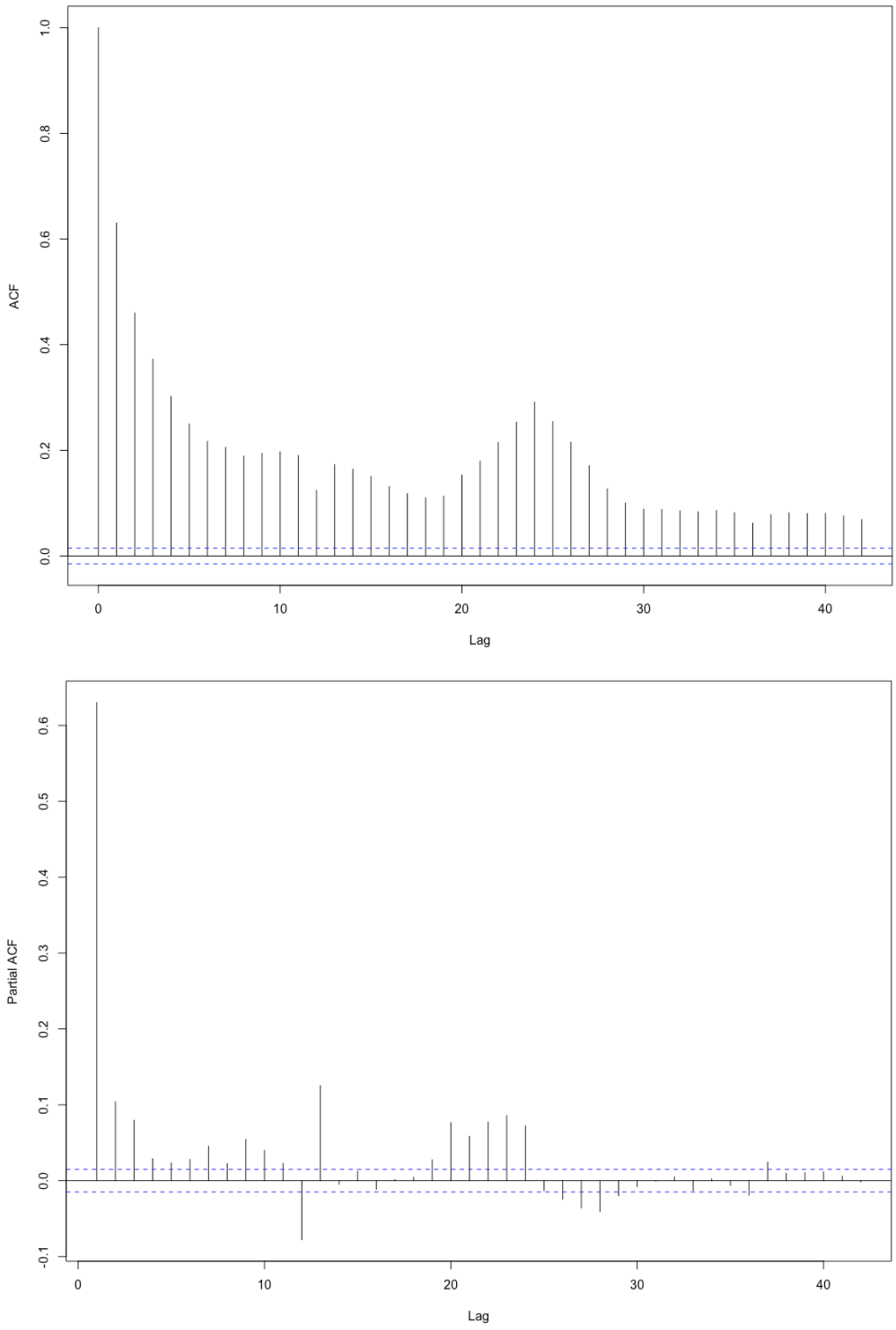


Figure 4.6 ACF and PACF for CO

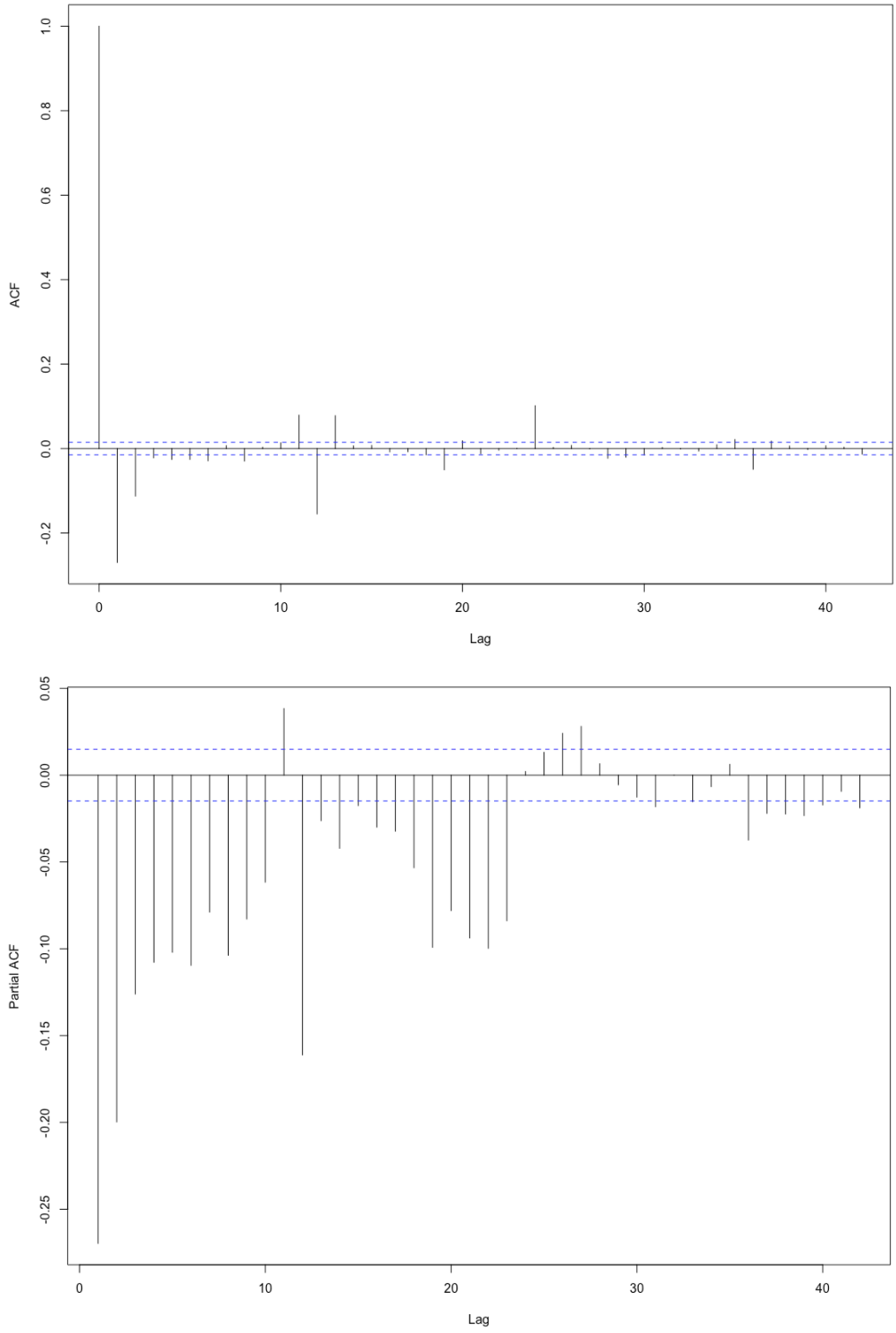


Figure 4.7 ACF and PACF for first order of CO

ACF plotting in Figure 4.7 suggests a first or second order of MA process, whereas PACF indicates many possibilities of order for AR process ranging from 1st to 10th order. For ACF, the past contribution to the process happens in the first and second lag with the respective factors of -0.269 and -0.112. For PACF, prior values starting from lag 1 to 10 have

negative contribution by the respective factors of: 0.27, 0.2, 0.13, 0.11, 0.10, 0.11, 0.79, 0.1, 0.83, and 0.62. Table 4.2 shows the result of various configurations.

p	d	q	P	D	Q	AIC
1	1	1				27199
1	0	0				29418
0	0	1				37669
5	1	4				26886
0	1	5				27233
2	1	2	2	1	2	27233
2	1	2	2	0	2	27064 *
1	1	1	1	1	1	27635 *
5	1	4	1	0	1	26807
5	1	4	1	1	1	27005
1	1	1	12	0	12	26346
1	1	1	24	0	24	**
1	1	1	12	1	12	26369
1	1	1	24	1	24	25805*
0	1	5	12	1	12	26313 *
4	1	5	3	0	3	26696

Table 4.2 Diagnostic checking for CO models using AIC.

Based on Table 4.2, the models for CO are: (0,1,5) x (12,1,12), (1,1,1) x (12,1,12), (1,1,1) x (12,0,12), and (1,1,1) x (24,1,24).

4.4.3 Nitrogen Dioxide

Two-years of measurement of nitrogen dioxide (NO_2) is shown in Figure 4.8. The NO_2 readings on the site are incomplete. There are 3,334 missing hours in the data, where 51 missing hours occurred in 2015. Another 3283 hours were missed in 2014. The missing data is noticeable in Figure 4.8. The missing periods are:

- 18th April 2014 9:00 AM to 22nd April 2014 at 12:00 PM,
- 1st May 2014 1:00 AM until 9th September 2014 at 12:00 PM
- 13th, 19th, and 30th September 2014.
- 1st, 3rd, 16th, 25th, 29th October, and
- 25th November, and
- 11th December 2014.

The two-year NO_2 concentration average is 34.44 ppb with 0.189 ppb and 132.4 ppb as the minimum and maximum values, respectively. The standard deviation is 28.56 ppb and variance is 344.39 ppb.

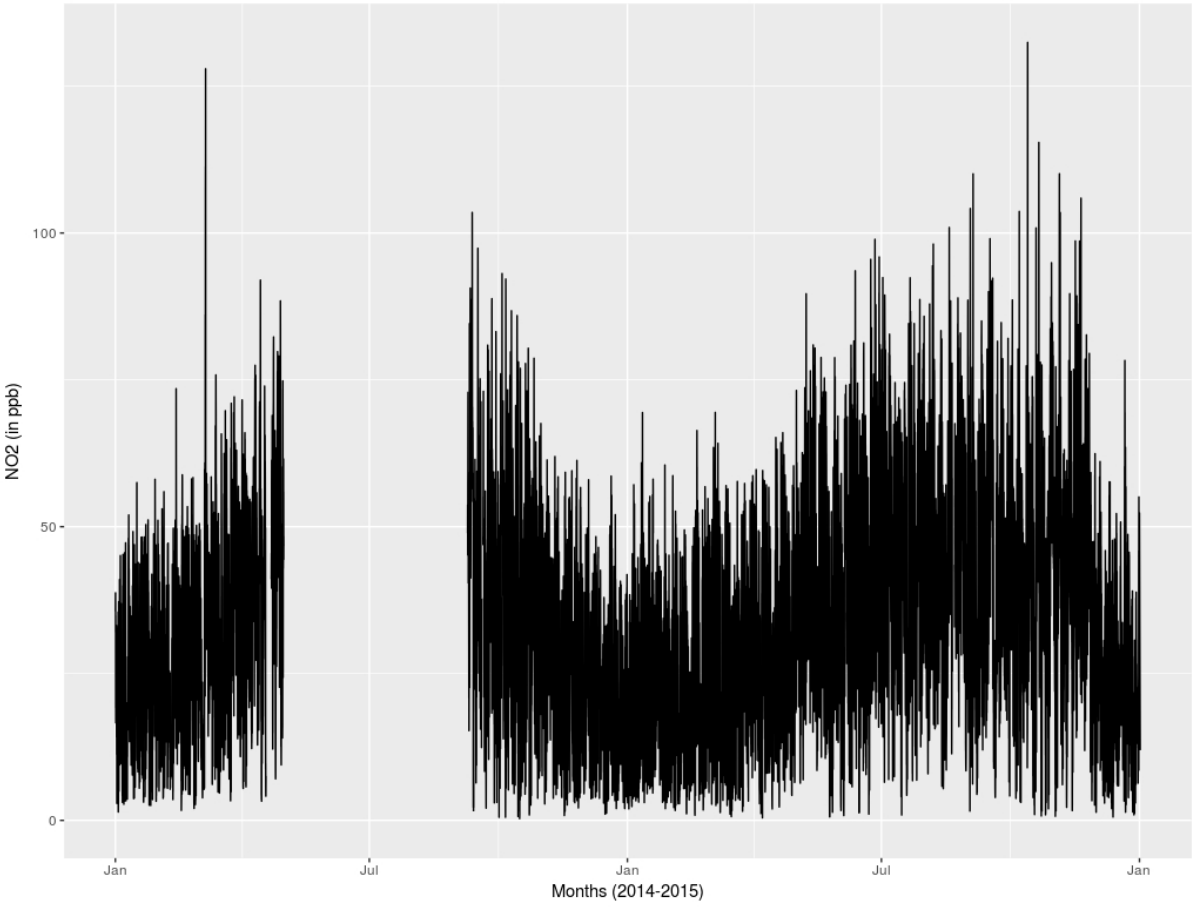
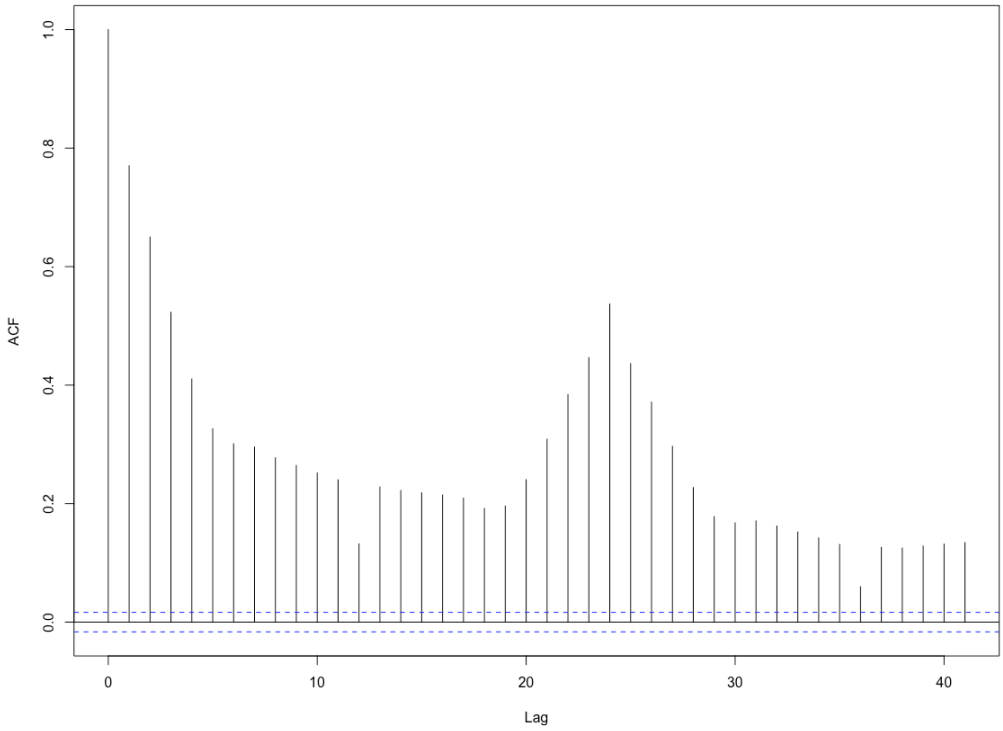


Figure 4.8 The plotting of nitrogen dioxide between 2014 and 2015

ACF and PACF are plotted, in Figure 4.9 below.



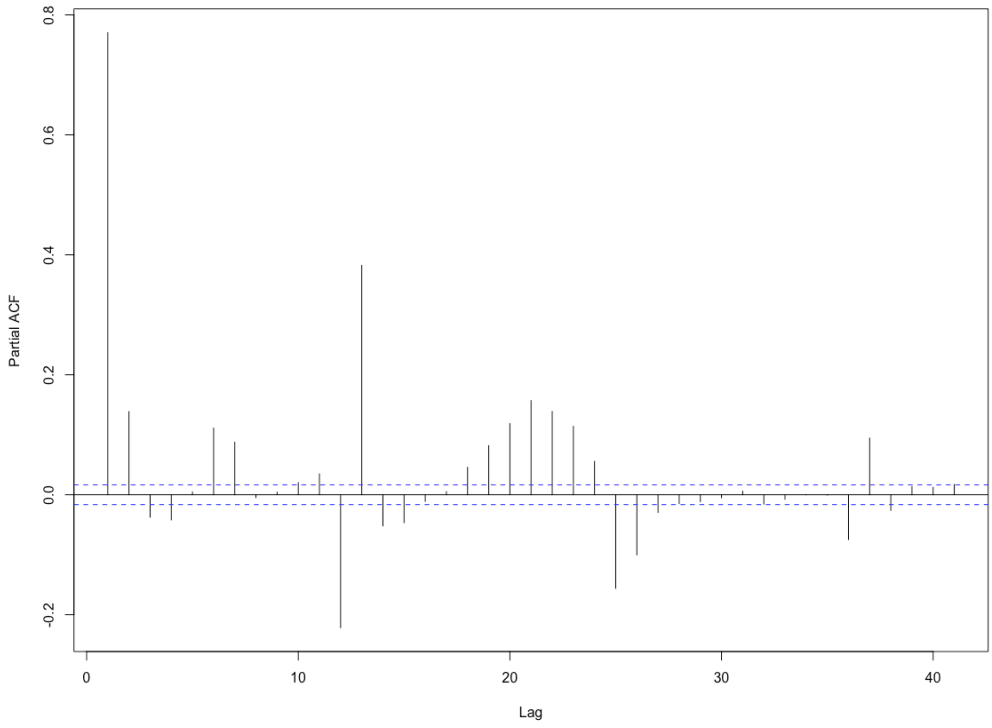


Figure 4.9 ACF and PACF of NO_2

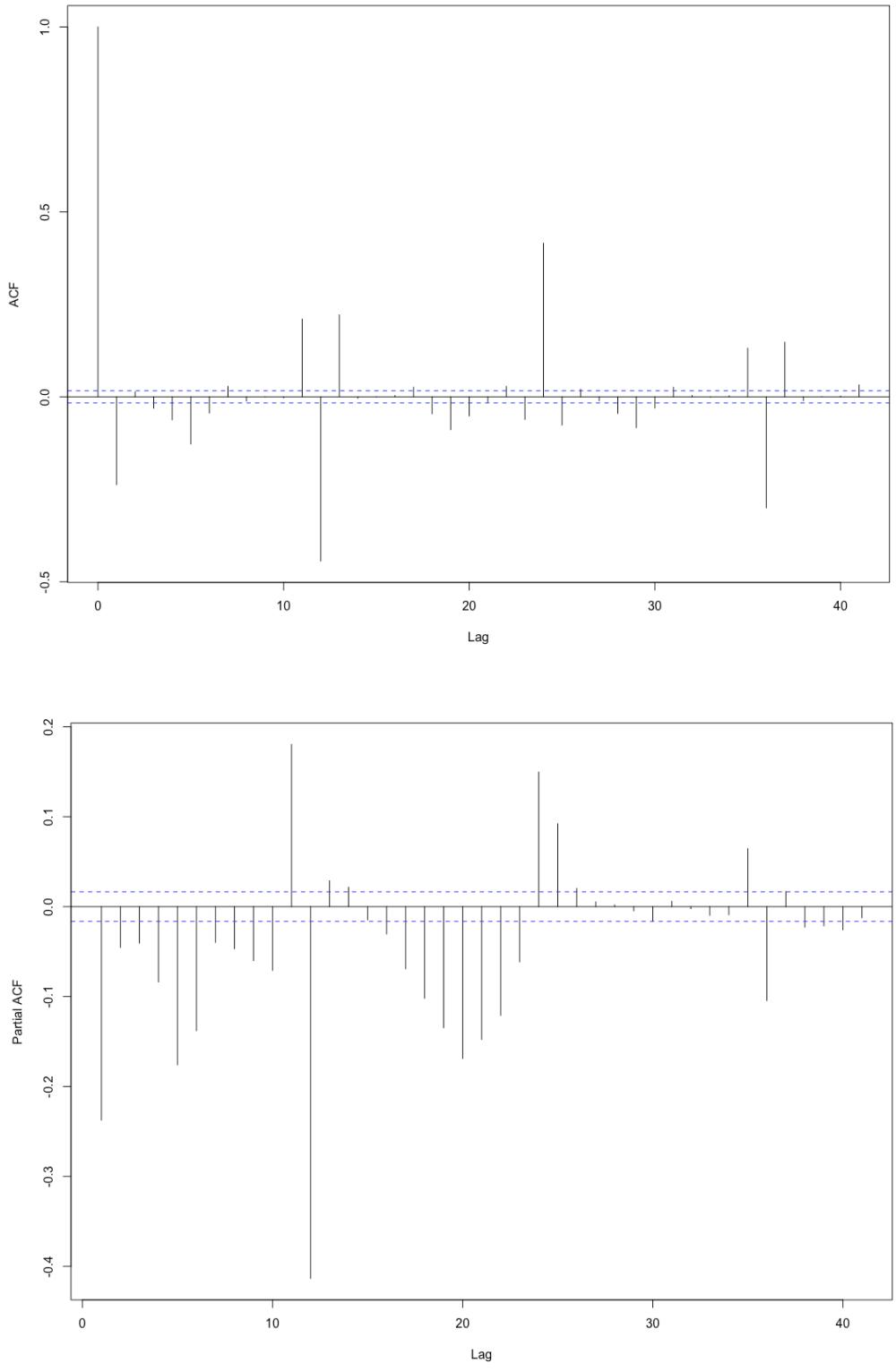


Figure 4.10 One order of differencing of ACF and PACF for NO_2

ACF in Figure 4.9 shows an indication of non-stationary process as the lags slowly decay. One order of differencing applied to NO_2 data and is illustrated in Figure 4.10. The first

lag has an impact to the series in the ACF plot. The seasonal ARIMA model might also be considered in the model, particularly at lag 12 and 24.

Following initial assessment of the ACF and PACF plots, AIC values are calculated for various configurations of lags, results given in Table 4.3.

p	d	q	P	D	Q	AIC
1	1	1				109,918
1	0	0				111,699
0	0	1				131,324
5	1	4				108,418
0	1	5				109,689
2	1	2	2	1	2	109,669 *
2	1	2	2	0	2	109,735
1	1	1	1	1	1	110,715
5	1	4	1	0	1	108,051
5	1	4	1	1	1	108,425
1	1	1	12	0	12	104,665
1	1	1	24	0	24	104,665
1	1	1	12	1	12	104,552
1	1	1	24	1	24	102,373
0	1	5	12	1	12	104,506 *

Table 4.3 Diagnostic checking for NO_2 models using AIC

From Table 4.3 the three lowest AIC values are (1,1,1) x (24,1,24), (1,1,1) x (12,1,12), and (0,1,5) x (12,1,12).

4.4.4 Humidity

The reading of humidity per hour in the Riccarton Road site in the period of 2014 and 2015 is shown in Figure 4.11. The figure is likely to show a seasonal trend and a weakly stationary process. The average hourly humidity reading over two years is 68.96%, within the range of 11.97% and 98.98%. The standard deviation is 18.05%. There was a great variety in the hourly reading of 325.99%.

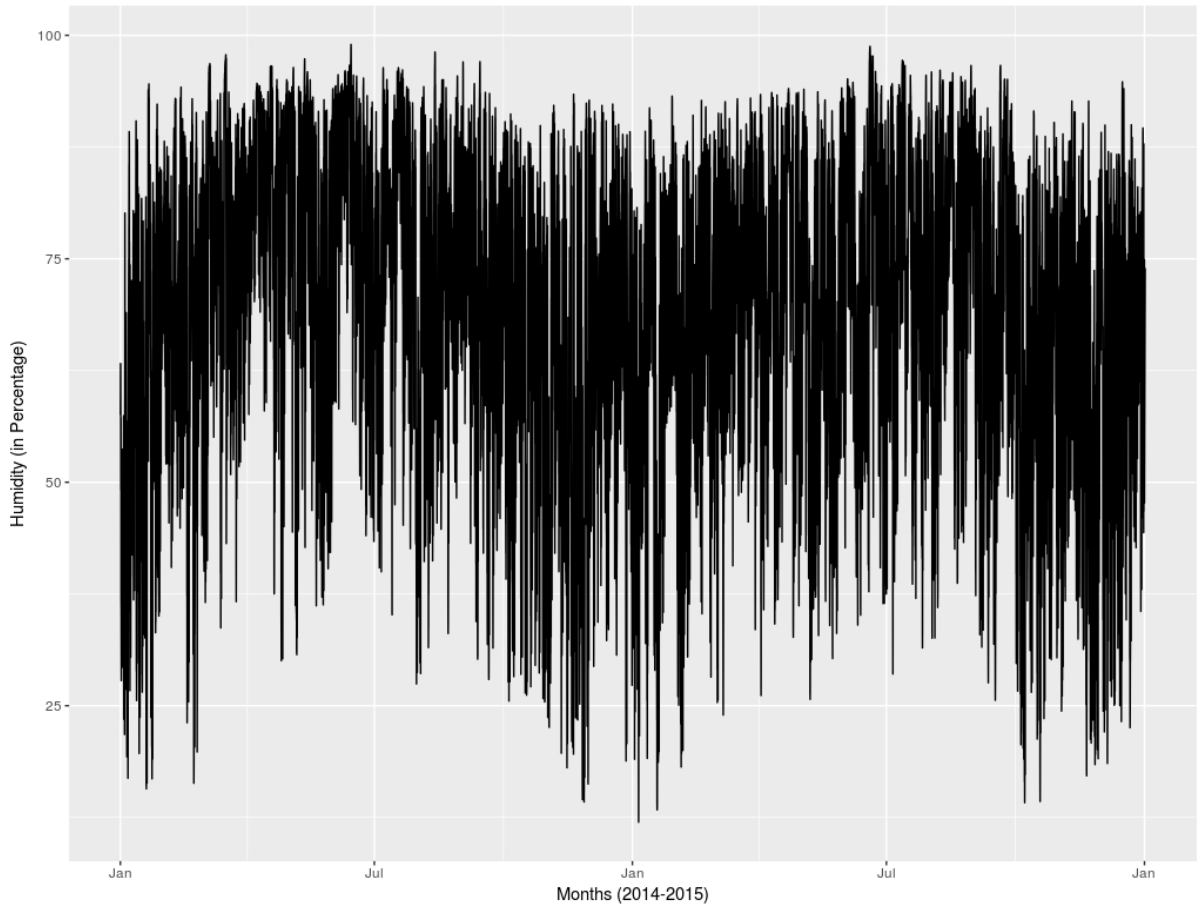


Figure 4.11 The hourly humidity on the Riccarton road site taken during the period of 2014 and 2015.

The plot of ACF and PACF for humidity is shown in Figure 4.12. Although this plotting of two-year humidity is likely to be a weakly stationary process, the ACF indicates the need for differencing.

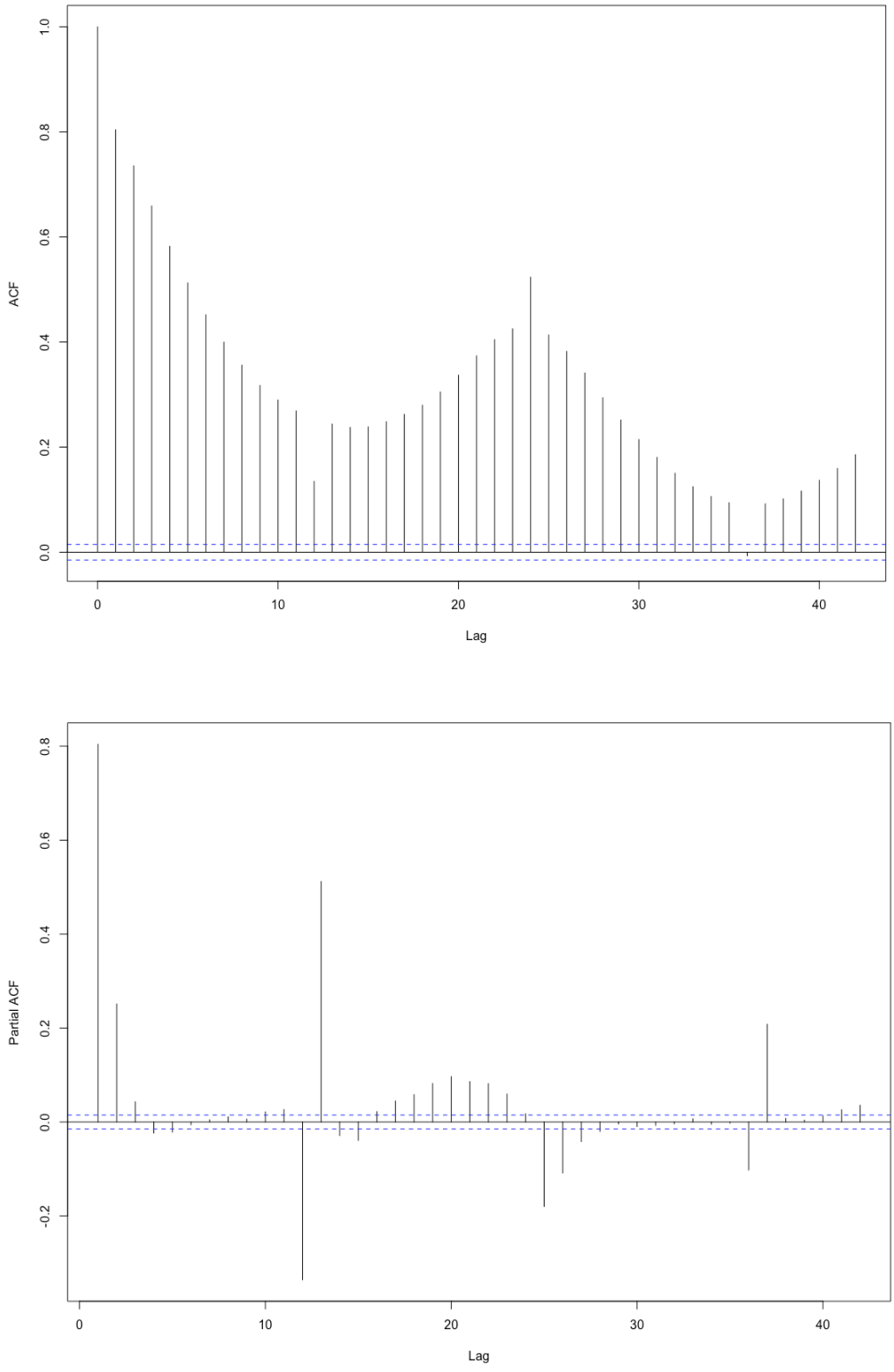
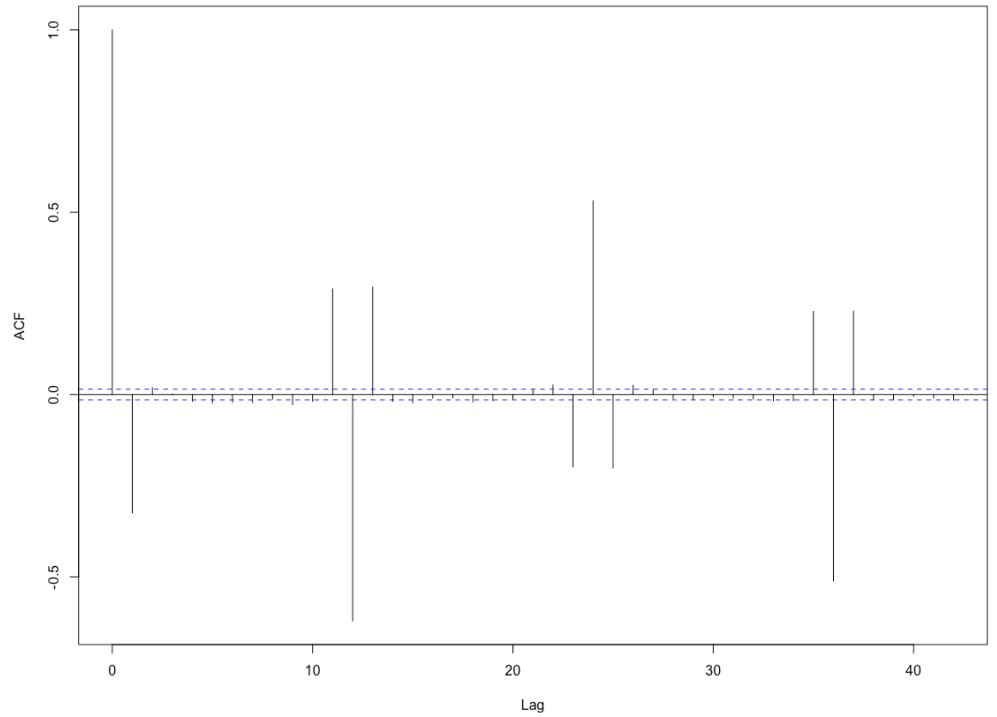


Figure 4.12 ACF and PACF for humidity



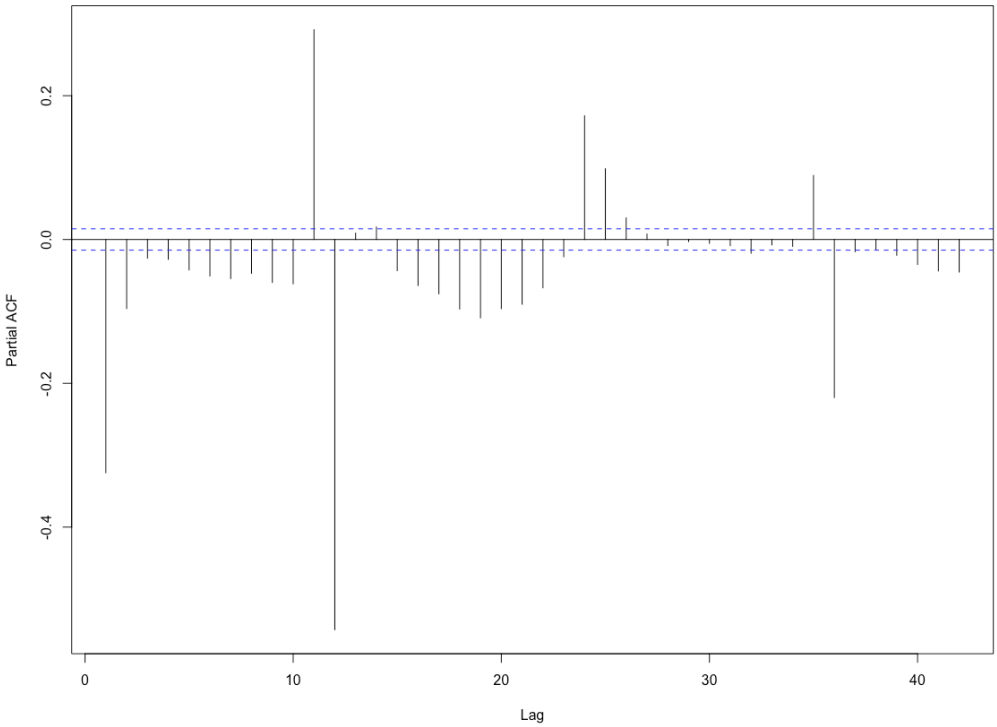


Figure 4.13 ACF and PACF for first order of humidity

Incomplete rows should be removed from the data before the difference process. The result is then plotted in Figure 4.13. Lag 1 and 13 appear to have a significant contribution to

the process, with respective factors of -0.324 and -0.299. Looking at the possibility to consider building a seasonal model, lag 12 and 24 have significant impact on the process with factors of -0.62 and 0.531, respectively. Also, it is interesting to consider lag 36 because its factor is -0.511 which shows an impact on the process as well. From Figure 4.13, the different value of orders is examined. The result is given in Table 4.4.

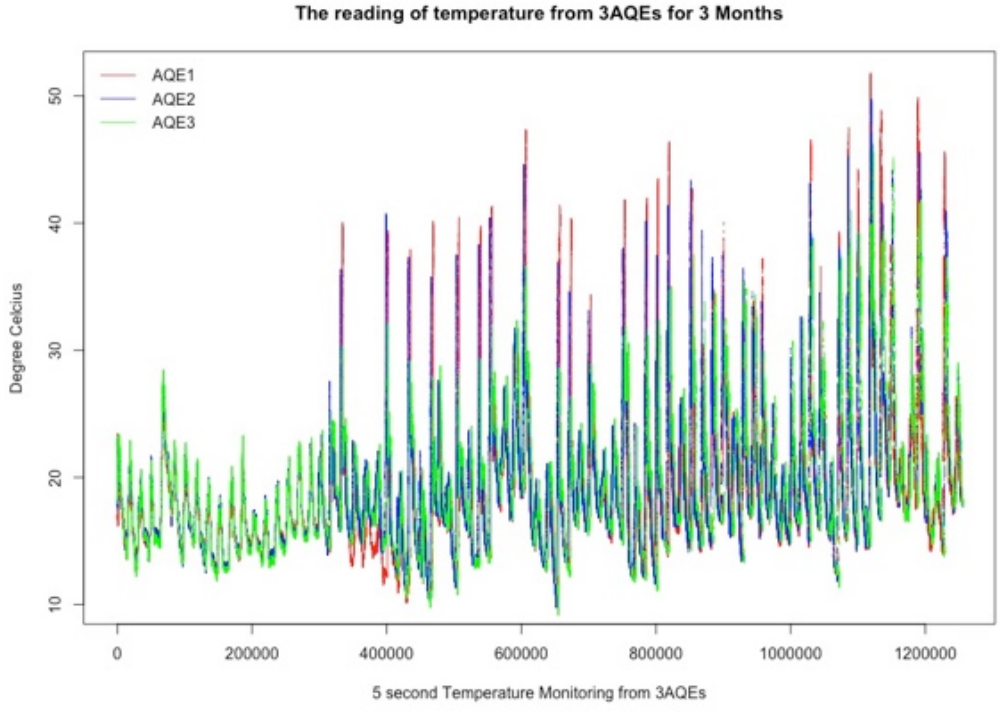
p	d	q	P	D	Q	AIC
1	1	1				132,534
1	0	0				134,570
0	0	1				178,511
5	1	4				130,443
0	1	5				132,309
2	1	3				131,472
2	1	3	12	1	12	122,025 *
2	1	2	2	1	2	131,501
2	1	2	2	0	2	131,482 *
1	1	1	1	1	1	133,951 *
5	1	4	1	0	1	130,445
5	1	4	1	1	1	130,457
1	1	1	12	0	12	130,457
1	1	1	24	0	24	**
1	1	1	12	1	12	122,025
1	1	1	24	1	24	**
0	1	1	12	1	12	122,228

Table 4.4 Diagnostic checking for humidity models using AIC

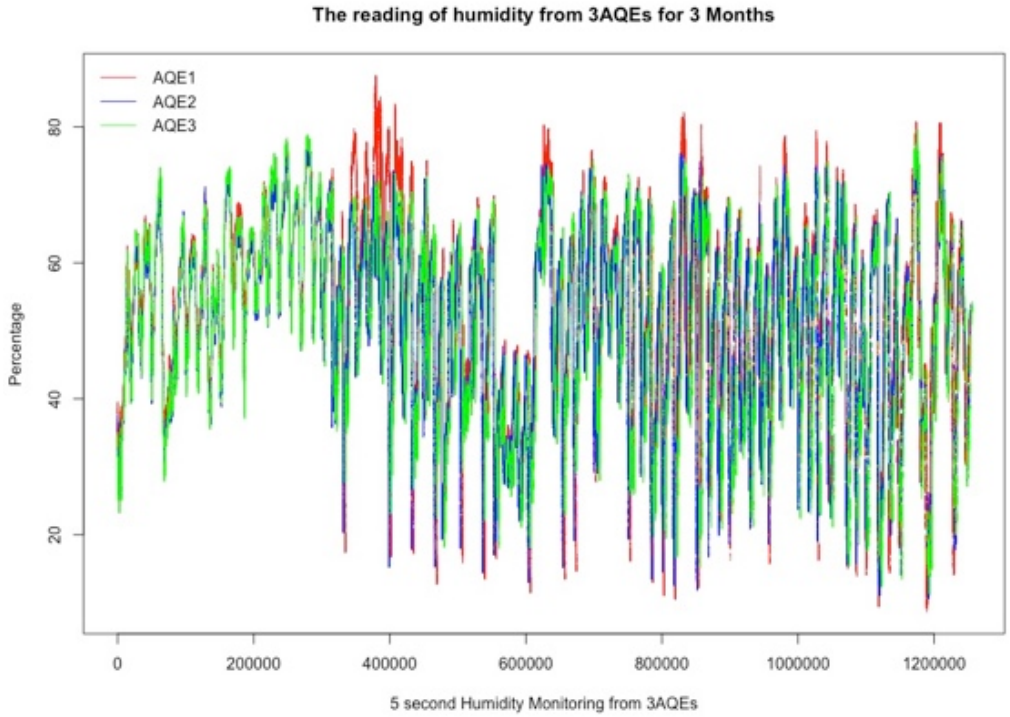
Looking at the AIC values in Table 4.4, two models are chosen: (0,1,1) x (12,1,12), (1,1,1) x (12,1,12), and (2,1,3) x (12,1,12).

4.5 Difference Among AQE Sensors

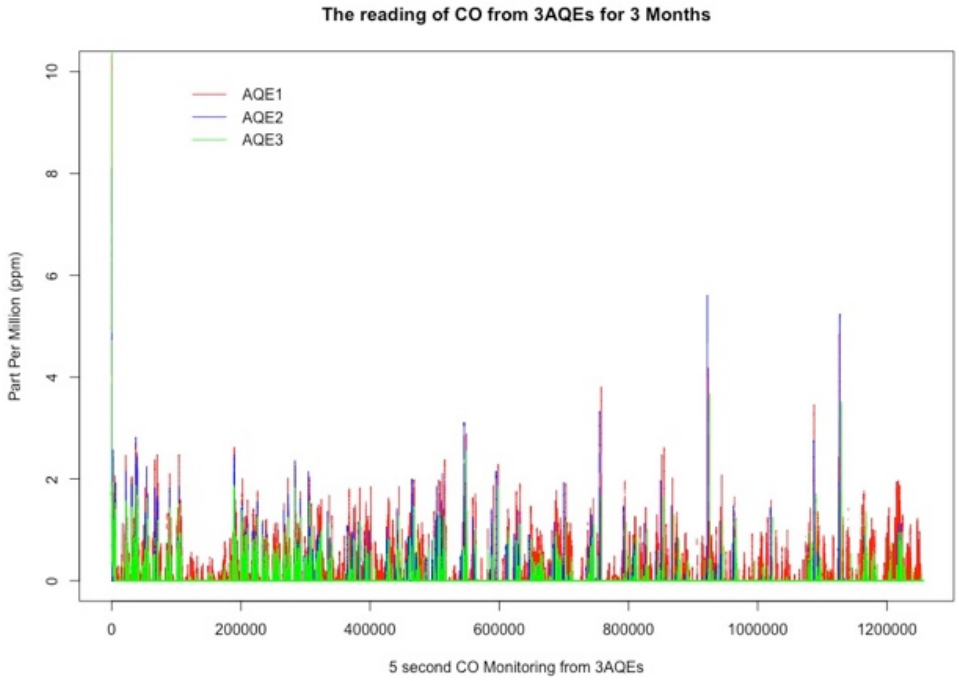
The AQE readings are now plotted and analysed statistically. The output of the 5-second reading of AQE from 4 sensors was plotted in Figure 4.14.


The reading of temperature from 3AQEs for 3 Months
AQE1
AQE2
AQE3
Degree Celcius
10
20
30
40
50
0
200000
400000
600000
800000
1000000
1200000
5 second Temperature Monitoring from 3AQEs

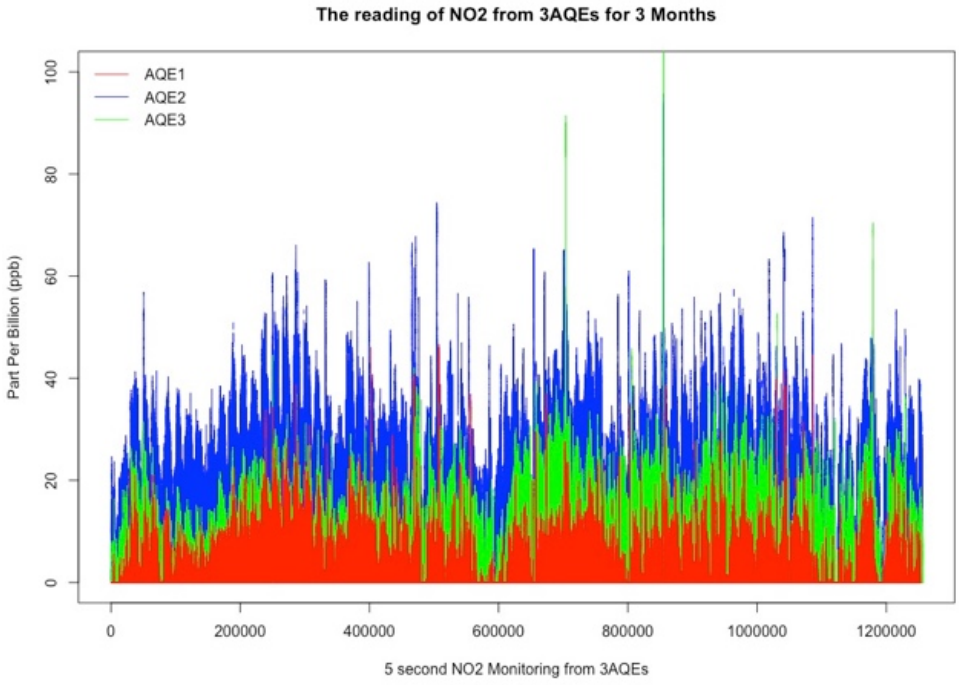
(a)


The reading of humidity from 3AQEs for 3 Months
AQE1
AQE2
AQE3
Percentage
20
40
60
80
0
200000
400000
600000
800000
1000000
1200000
5 second Humidity Monitoring from 3AQEs

(b)



(c)



(d)

Figure 4.14 5-second reading of three Air Quality Eggs in Riccarton road during a period of three months from the sensors:(a) carbon monoxide (CO). (b) nitrogen dioxide (NO_2). (c) temperature. (d) humidity.

The index of agreement (*d*-value) is used to assess the agreement reading between the corresponding sensors using the *d()* function from the hydroGOF library [83]. The library

is specifically used for graphical comparison and has a function to perform index of agreement. The index of agreement of '0' indicates no agreement between the sensors, while '1' indicates a perfect agreement. Further detail of the index of agreement is explained in Section 5.3.2. Table 4.5 shows the *d*-value of each sensor type. The table indicates temperature, humidity and CO sensors have the same agreement on the reading (more than 0.8), while NO_2 sensors have a poor level of agreement.

Type of Sensors	Index of Agreement Among Sensors
Temperature	0.98
Humidity	0.98
CO	0.80
NO_2	0.40

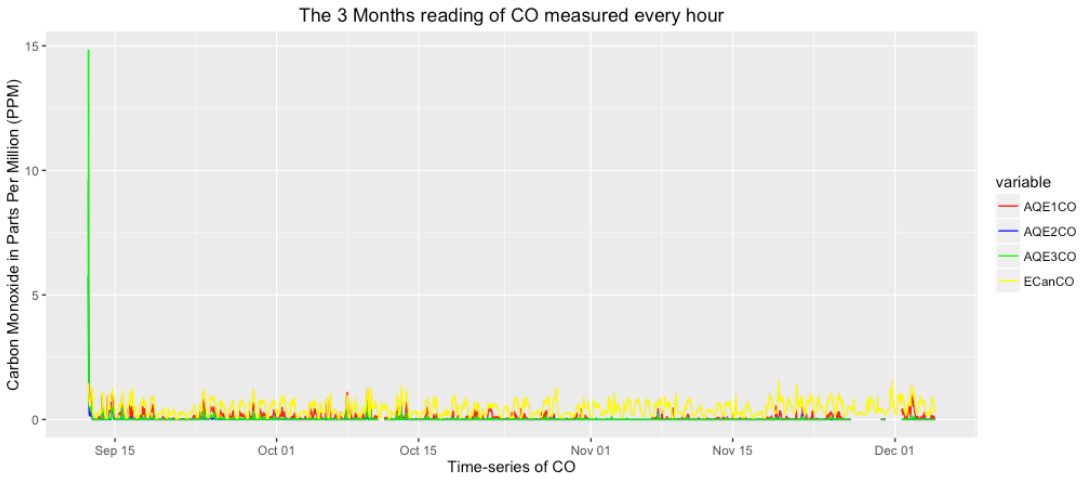
Table 4.5 Index of agreement between sensors on AQEs

The AQEs were averaged into hour long readings where the missing hours were skipped. The AQE and ECan data were compared visually, as plotted in Figure 4.15. The readings difference between hourly ECan and hourly AQE data were compared using an analysis of variance with the *aov* function [84] in Table 4.6. The sum of squares indicates the total variation attributed to a factor. In general, the readings of AQE1 sensors diverge more from the ECan in Table 4.6. AQE3 sensors have the smallest total difference in readings to ECan, compared to the other two AQEs. There is an exception to the AQE3's humidity sensor because AQE2's humidity sensor total variation was the smallest. Regarding the CO sensors, the total variation of AQE1 is 21 ppm, AQE2 deviates by 16.33 ppm, and 2.31 ppm for AQE3 from the ECan mean in Table 4.6. Similarly, the total difference of AQE1, AQE2, and AQE3 were 21,769.3, 4,654.2, and 1,727 ppb, respectively, in comparison to the ECan mean. The humidity sensors on the AQEs show the biggest variation in comparison to ECan, particularly for AQE1 by 226,381.5 %.

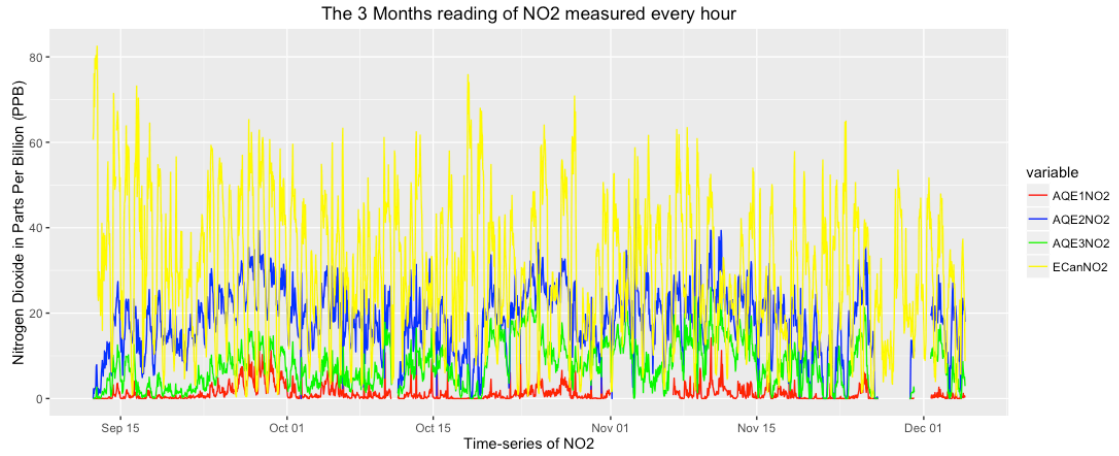
Sum of Squares AQE1	Sum of Squares AQE2	Sum of Squares AQE3	Residuals	Sensor Types
21.00 ppm	16.33 ppm	2.31 ppm	96.31 ppm	CO
21,769.3 ppb	4,654.2 ppb	1,727.0 ppb	466,703.8 ppb	NO_2
8,344.02 ^{0}C	1,045.72 ^{0}C	1,025.11 ^{0}C	18,865.01 ^{0}C	Temperature
226,381.5 %	2,786.8 %	3,009.5 %	343,102.3 %	Humidity

Table 4.6 Analysis of variance: comparing readings difference between individual AQEs and ECan

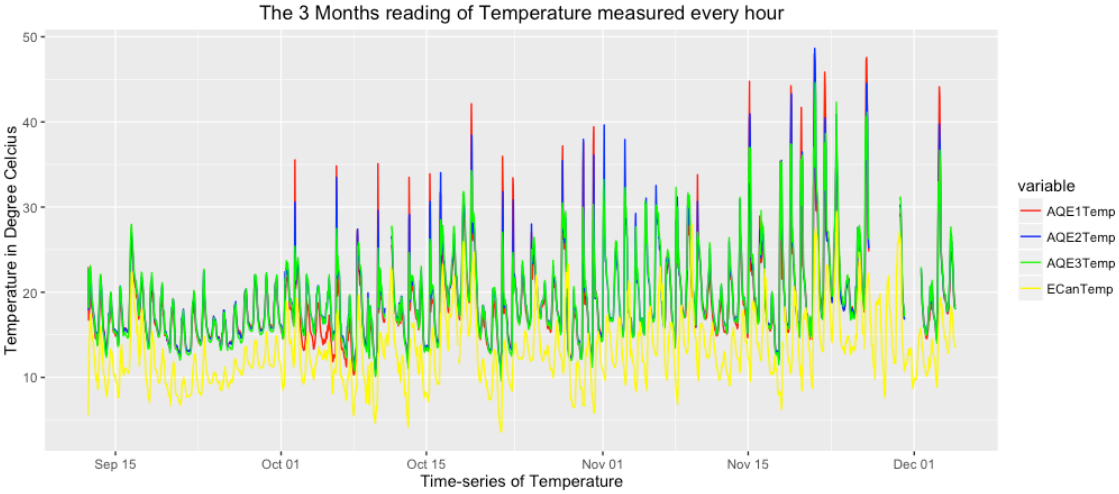
Apart from the early measurement when the AQEs were turned on for the first time, the CO reading from all AQEs and ECan during the experiment period never reached 2.5 ppm, as illustrated by Figure 4.15a. The AQEs' CO sensors underestimate the ECan value. Similar readings occurred in the NO_2 readings (Figure 4.15b) and humidity (Figure 4.15d) where the AQEs readings were lower than ECan. Meanwhile, the temperature readings overestimate the ECan readings on the site in Figure 4.15c.



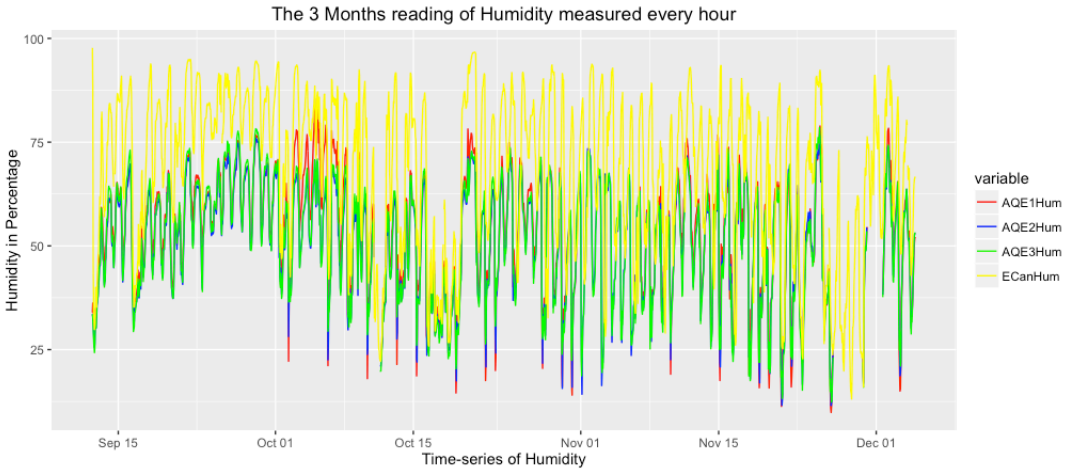
(a)



(b)



(c)



(d)

Figure 4.15 Hourly reading of three AQEs and ECan in Riccarton road during a period of 3 months from the sensors: (a) CO (b) NO_2 (c) temperature (d) humidity.

We have now established a null hypothesis: there is no mean difference between all the corresponding sensors on AQEs, while the alternative hypothesis is the opposite of the null hypothesis. The null hypothesis (H_0) and the alternative hypothesis (H_1) can be written mathematically as:

$$H_0: \mu_1 = \mu_2 = \mu_3$$

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

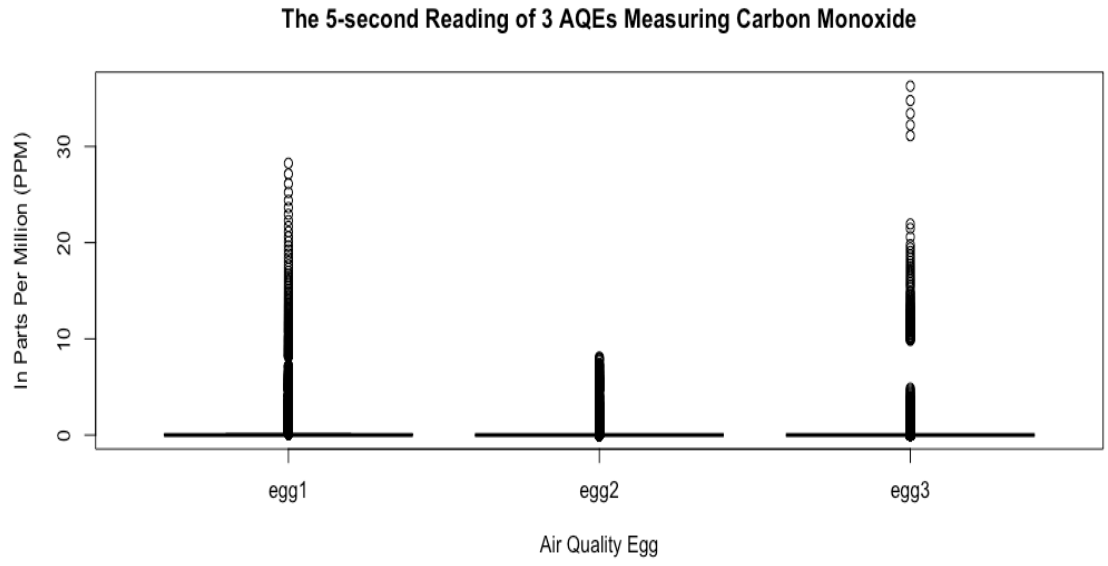
Take a null hypothesis for CO sensors as an example, the null hypothesis equation above suggests that the mean of AQE1 (μ_1), AQE2 (μ_2), and AQE3 (μ_3) are all equal. The one-way analysis of variance (ANOVA) and the Tukey test are chosen to test the null hypothesis [85]. ANOVA is a null hypothesis test to determine group means by partitioning variance. Only one independent variable is considered in the ANOVA test, hence it calls as one-way. The F-test using ANOVA determines the relationship between multiple predictors in predicting the response so it can validate whether accepting or rejecting a null hypothesis. A null hypothesis can be rejected if the F ratio is statistically significant, depicting that at least one of the group means is different from the others, but we cannot determine which group it is. A test between any pair of groups can be carried out to further determine the difference among them. This is called post hoc or posteriori testing.

Box-and-whisker plots were used to summarize the 5-second AQEs readings. A Box-and-whisker plot presents the whole data as illustrated in Figure 4.16a, Figure 4.17a, Figure 4.18a, and Figure 4.19a below. On these four figures, a box indicates the inter-quartile range, starting from the 25th to the 75th percentile. A horizontal line inside the box indicates the median. Dotted vertical lines with horizontal lines at the ends show the minimum and maximum values outside the inter-quartile range.

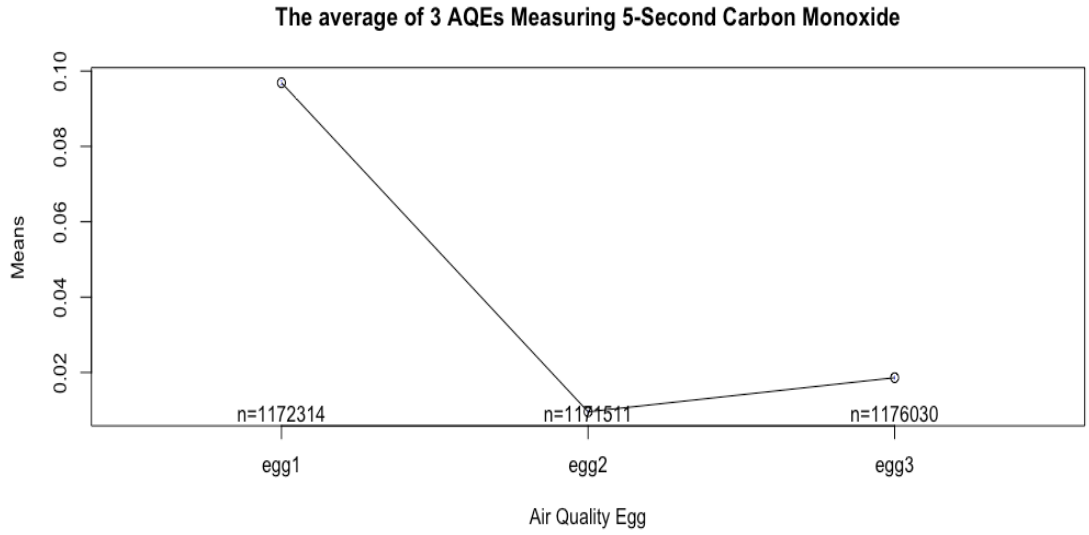
The results of investigating the null hypothesis are discussed in the next four sections below.

4.5.1 Carbon Monoxide

The 5-second CO monitoring sensors on the AQEs during the period of experiment were compared and statistical properties are illustrated in Figure 4.16a and Figure 4.16b. The box-and-whisker plot from Figure 4.16a seems to be unnoticed because the inter-quartile range was narrow. Instead, there were a lot of scattered readings away from the inter-quartile range. Egg3 (AQE3) showed the biggest variance (more than 30 ppm), while the smallest variance occurred at egg2 (AQE2). Looking at the mean from Figure 4.16b, it implied that the average of the three CO sensors was relatively small, less than 0.1.



(a)



(b)

Figure 4.16 5-second CO reading from three AQEs. (a) box-and-whisker plot. (b) comparison of means (n=total number of data) of all AQEs.

Next, a one-way analysis of variance (ANOVA) test was conducted using R, and the table below was obtained.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Egg	2	5408	2704	75908	<2e-16	***
Residuals	3519852	126746	0			
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1						

Table 4.7 ANOVA result for CO on AQEs

From Table 4.7, the following information can be extracted: F (2, 3519852) = 75,908, sum-of-square error = 5,408, mean square error = 2,704, and p value is less than 2 x 10^{-16}. Given the two ranges of the degree of freedom (*Df*) between 2 and 3,519,852, the critical F-value was obtained and its values within the range of 0.03 and 2.59 for 2.5% and 92.5% bands. The F-value extracted from Table 4.7 (75,908) is too far from the critical values. Table 4.7 indicates that the means of AQEs were all different. Therefore, the null hypothesis is rejected, and the alternative hypothesis is accepted.

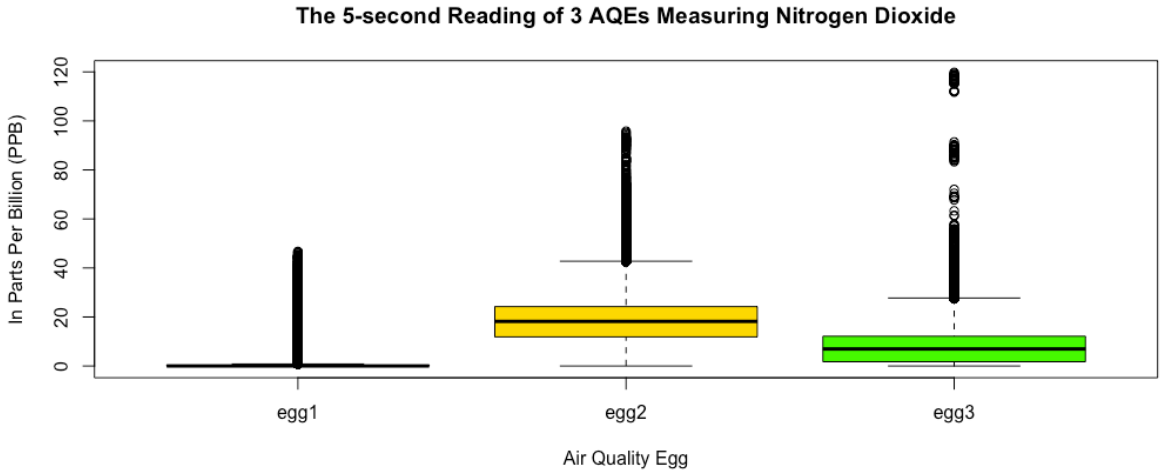
Table 4.7 also suggests a very low p-value. A post hoc test was conducted to determine the differences among AQEs. Of the post hoc tests, Tukey's honest significant difference (HSD) post hoc test was chosen [85]. Brown suggested that Tukey's method was suitable for general use [86]. Table 4.8 depicts the result from R.

Comparison	diff	lwr	upr	p adj
egg2 - egg1	-0.087335102	- 0.087916100	- 0.086754104	0
egg3 - egg1	-0.078281008	- 0.078861447	- 0.077700569	0
egg3 - egg2	0.009054095	0.008473556	0.009634633	0

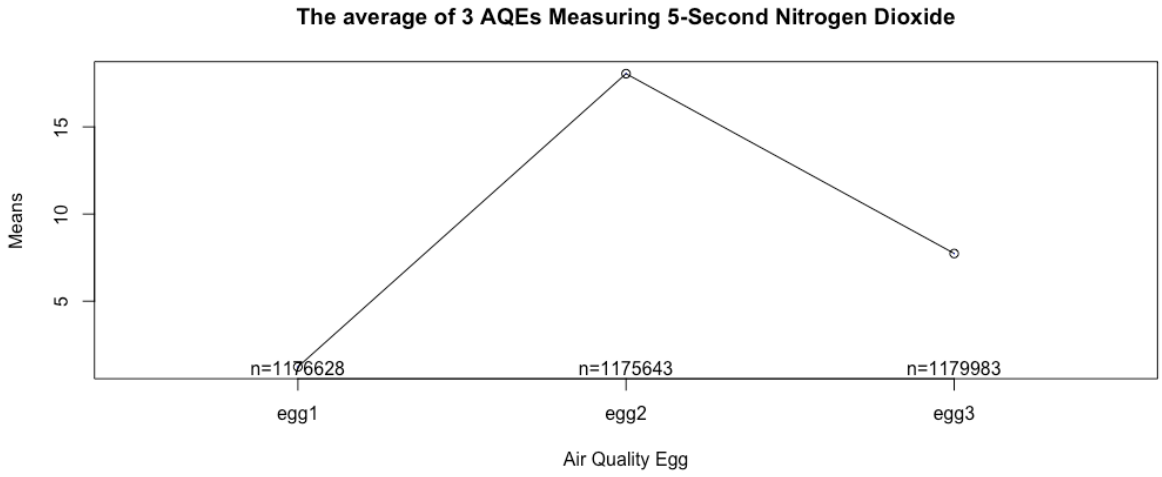
Table 4.8 Tukey post hoc test on the 5-second reading of CO on AQEs

The ANOVA test confirmed there is a difference between AQEs, but was not able to quantify the difference among the AQEs. Column *diff* and *p adj in* Table 4.8 shows there was significant difference (p adj < 0.001) between AQEs in measuring carbon monoxide where the biggest means difference occurs between egg1 and egg2 by 0.087 ppm or 87 ppb. Note that the *p adj* (or *p-value*) column in Table 4.8 (and also Table 4.10, Table 4.12, Table 4.14) indicates a value of zero. It is likely that R rounded it to zero because p adj was nearly zero [87]. Table 4.8, Table 4.10, Table 4.12, and Table 4.14 shows the lower (lwr) and upper (upr) difference between two AQEs, while the diff column shows the average difference.

4.5.2 Nitrogen Dioxide



(a)



(b)

Figure 4.17 5-second NO_2 readings from three AQEs. (a) box-and-whisker plot. (b) comparison of means.

Figure 4.17a shows NO_2 readings on three AQE sensors using the box-and-whisker plot. The plot shows a relatively big difference of dispersed readings among the AQEs. It is interesting to note that the NO_2 sensor on AQE1 has a narrow range reading in Figure 4.17a. The maximum AQE1 NO_2 ambient measurement was 50 ppb. The AQE2 reading was from

zero to almost 100 ppb. The AQE3 readings range from zero up to 120 ppb. Figure 4.17b further describes the means difference between the three devices.

Table 4.9 describes analysis of variance of the 5-second nitrogen dioxide sensors on all AQEs. The table shows the following statistical properties: F (2, 3532251) = 1,782,262, *p-value* < .001, and large values of sum square error and mean square error. The F-value of 1,782,262 has exceeded the quantile band of F-distribution between 0.03 and 2.59. Therefore, Table 4.9 suggests that the alternative hypothesis was taken. Consequently, the means of NO_2 sensors on AQEs were likely to be different.

On the other hand, a very small *p-value* indicates that there was a significant difference among NO_2 sensors on AQEs. Table 4.10 shows the Tukey test between AQEs. The biggest means difference was between egg1 and egg2 by 16.82 ppb. Table 4.10 indicates all AQEs significantly differed from each other.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Egg	2	169260077	84630038	1782262	<2e-16	***
Residuals	3532251	167727564	47			
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1						

Table 4.9 ANOVA result for NO_2 on AQEs

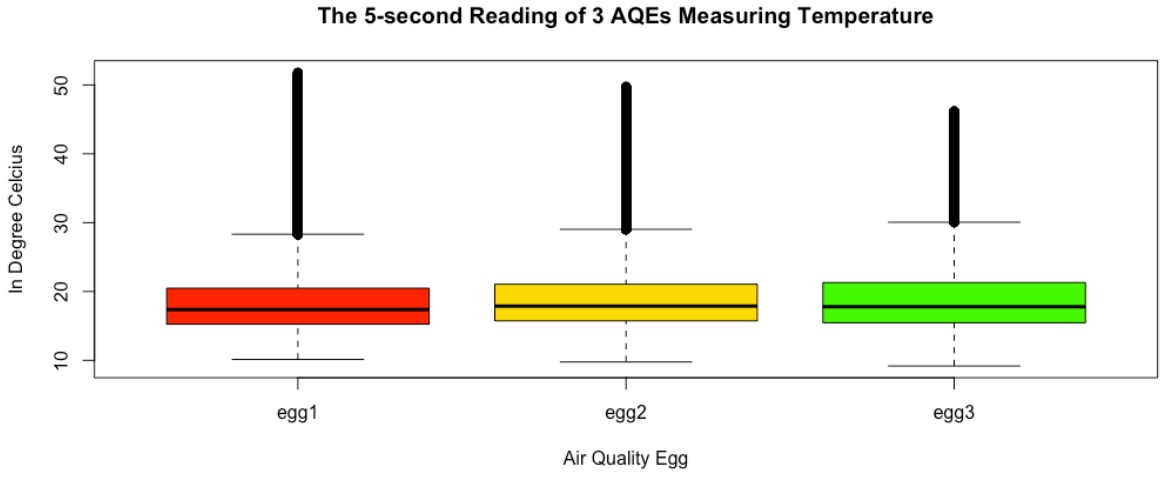
Comparison	diff	Lwr	upr	p adj
egg2-egg1	16.822176	16.801115	16.843236	0
egg3-egg1	6.504663	6.483622	6.525704	0
egg3-egg2	-10.317512	- 10.338558	-10.296467	0

Table 4.10 Tukey post hoc test on the 5-second reading of NO_2 on AQEs

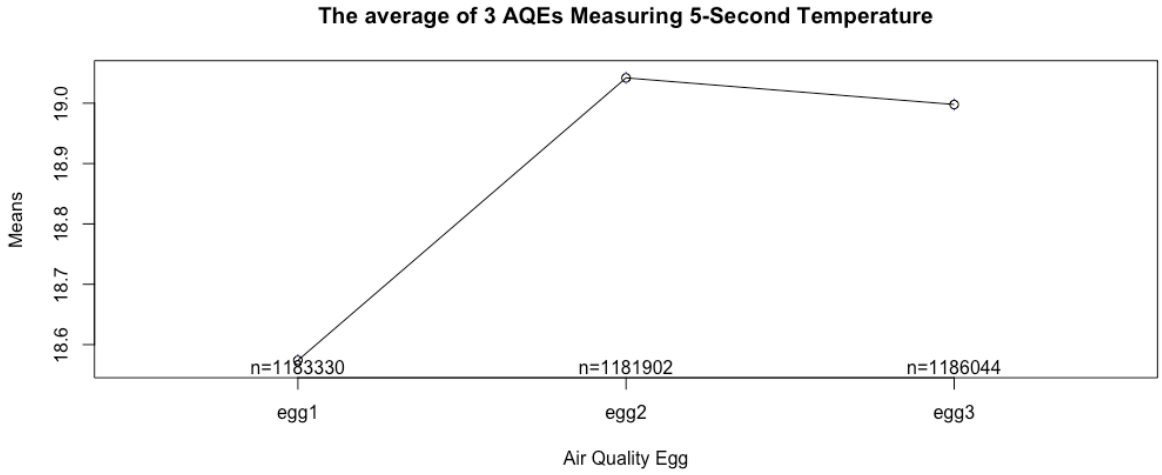
4.5.3 Temperature

The temperature readings on AQEs are likely to agree in Figure 4.18a. The average temperature on the three AQEs was roughly 19 degrees Celsius. Although the means values are seasonal, the two figures indicate that the temperature sensor readings have a few

variations. The Tukey test in Table 4.12 shows that F (2, 3551273) = 3,019, sum-of-square error = 157,969, mean square error = 78,984, and *p-value* is less than 2 x 10^{-16}. The quantile F-distribution in the range of 0.03 (2.5%) and 2.59 (92.5%) indicates that the alternative hypothesis was accepted and that there is a significant mean difference between AQEs as indicating by a low *p-value*. The Tukey test depicts that the three AQEs significantly differed from each other. However, the difference between AQEs is relatively small with a minimum of 0.03 degree Celsius and maximum of 0.48 degree Celsius in Table 4.12.



(a)



(b)

Figure 4.18 5-second temperature reading from three AQEs. (a) box-and-whiskey plot. (b) comparison of means.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Egg	2	157969	78984	3019	<2e-16	***
Residuals	3551273	92915206	26			
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1						

Table 4.11 ANOVA result for temperature on AQEs

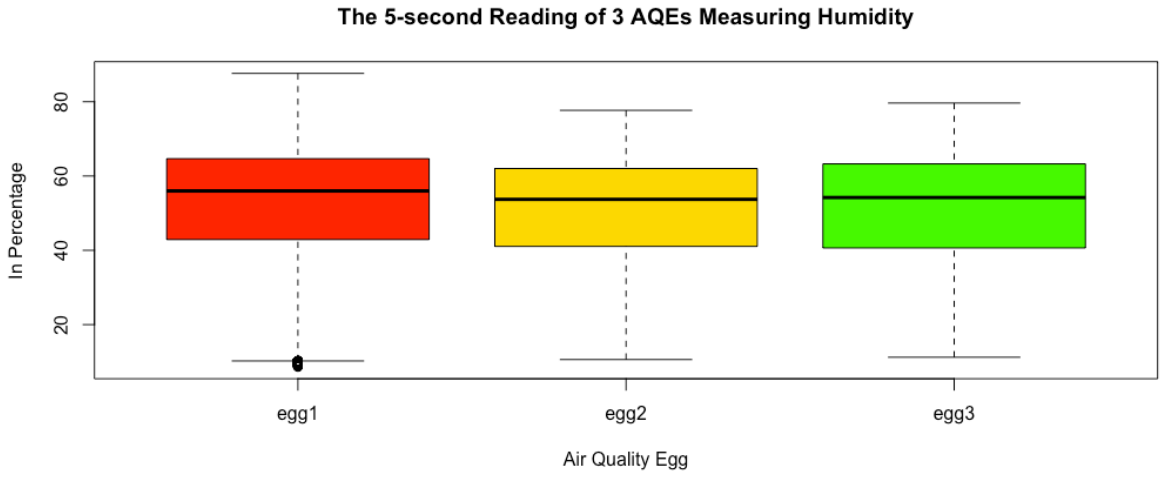
Comparison	diff	lwr	Upr	p adj
egg2-egg1	0.46791132	0.45232130	0.48350134	0
egg3-egg1	0.42378337	0.40820698	0.43935977	0
egg3-egg2	-0.04412794	-0.05970905	-0.02854684	0

Table 4.12 Tukey post hoc test on the 5-second reading of temperature on AQEs

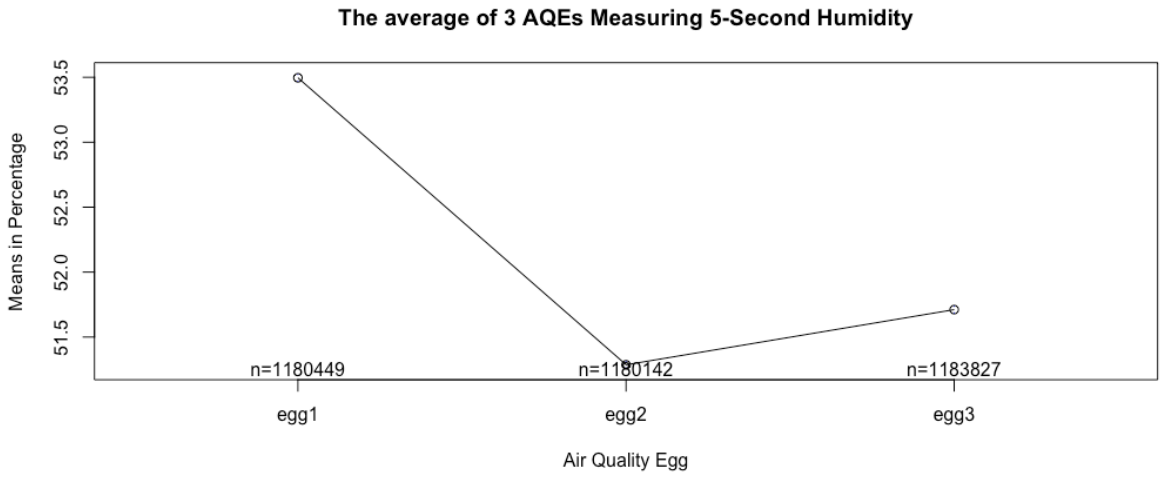
4.5.4 Humidity

Similar to the temperature readings, the humidity readings on three AQEs had similar means, shown in Figure 4.19a. The figure suggests that the humidity sensors' readings were likely different by more than 2.2%. The ANOVA test in Table 4.13 shows that F (2, 3544415) = 8,124, sum-of-square error= 3,246,287, mean-square error= 1,623,144, and *p-value* nearly zero (< $2x10^{-16}$). Although the three eggs appear to have similar means, the ANOVA test proves the opposite. Because the *F-value* was out of the quantile F-Distribution (0.03 to 2.59),

it suggests that all AQEs were all different, as the null hypothesis was rejected. However, because the *p-value* was nearly zero, the Tukey test shows there was a significant difference in the reading of humidity among AQEs. A relatively small difference between AQE1, AQE2, and AQE3 is shown in Table 4.14 where the biggest variation occurred between egg1 and egg2 by 2.2%, while the smallest one on AQE2 and AQE3 is 0.4%.



(a)



(b)

Figure 4.19 5-second humidity reading from three AQEs. (a) comparison of its reading. (b) comparison of means.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Egg	2	3246287	1623144	8124	<2e-16	***
Residuals	3544415	708129109	200			
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1						

Table 4.13 ANOVA result for humidity on AQEs

Comparison	Diff	Lwr	upr	p adj
egg2-egg1	-2.2097060	- 2.2528287	- 2.1665833	0
egg3-egg1	-1.7848422	- 1.8279313	- 1.7417531	0
egg3-egg2	0.4248638	0.3817719	0.4679557	0

Table 4.14 Tukey post hoc test on the 5-second reading of humidity on AQEs

4.6 Testing Phase

The plotting of the AQE sensors in Section 4.5 shows there are fluctuations and sudden changes. The sudden change may be a result of environmental changes or a sensor fault. The availability of pre-defined data is required for evaluating the decision schemes in the testing phase. Zhang *et al* [56] noted that obtaining pre-defined data can be difficult, or even impossible, in the supervised learning approach. Although ECan data was used as a reference, the data has no pre-defined label of normal or outlier. To evaluate the accuracy of the four proposed decision schemes using supervised learning approach, the AQE data is assumed to have no outliers. This assumption is based on the fact that the AQEs were operated for the first time. So, the sensor reading is likely to be free from device faults. TP and FN can be calculated using this assumption, but FP and TN cannot be determined. So, FP and TN is set to 0 in the Equation 4.3.

The ARIMA estimators for CO, NO_2, temperature, and humidity sensors were obtained using past ECan data, as discussed in Section 4.4. These estimators are used only as a starting point because the result of an ARIMA estimator may vary and may depend on the data. Along

with these estimators, the analysis and evaluation stage involve assessing various configurations, such as extending the prediction range and different parameter estimation techniques. We assume the reference values would be on the normal distribution range of the ARIMA estimators' prediction. The prediction range using one or two standard deviations is tested in order to classify a reading as a normal one. The sensor reading is expected to be within one or two standard deviations away from the mean of the forecast value. One standard deviation assumes a value will be on the 68.2% band of the normal distribution, while two standard deviations expects a value will be within the 95.4% band of the normal distribution. The maximum likelihood (ML) and ordinary least squares (OLS) techniques were tested for estimating the parameter. There are 50, 60, 72, and 70 testing scenarios of monthly batches for temperature, humidity, CO, and NO_2, respectively. In terms of the weekly data sequence, there are 36, 54, 36, and 36 testing scenarios for temperature, humidity, CO, and NO_2, respectively. We chose non-seasonal ARIMA estimators for weekly data input because of similar AIC values in the training phase, following the suggestion of Sakamoto *et al* [81]. Another reason for picking the non-seasonal estimators is to reduce the computation power on the server. Obtained from the training phase in Section 4.4, the ARIMA estimators are:

(i). Monthly Temperature: (1,1,1) x (12,1,12), (1,1,1) x (24,1,24), (0,1,1) x (12,1,12), (1,1,1), and (0,1,1)

(ii). Monthly Humidity: (1,1,1) x (12,1,12), (0,1,1) x (12,1,12), (2,1,3) x (12,1,12), (1,1,1), (0,1,1), (2,1,3)

(iii). Monthly CO: (0,1,5) x (12,1,12), (1,1,1) x (12,1,12), (1,1,1) x (12,0,12), (1,1,1) x (24,1,24), (0,1,5), (1,1,1)

(iv). Monthly NO_2: (1,1,1) x (12,1,12), (1,1,1) x (24,1,24), (0,1,5) x (12,1,12), (0,1,5), (1,1,1)

(v). Weekly Temperature: (1,1,1), (0,1,1)

(vi). Weekly Humidity: (1,1,1), (2,1,3), (0,1,1)

(vii). Weekly CO: (0,1,5), (1,1,1)

(viii). Weekly NO_2: (0,1,5), (1,1,1)

ML was the default method for estimating parameters in the R ARIMA function. Dent suggested using ML to estimate parameters in two conditions: for general use, and for small or moderate sample sizes, while OLS was used for a larger dataset [75]. Parameter estimation can fail at the data analysis stage using ML for two reasons. First, if the calculation converges it can encounter a negative log-likelihood ending with infinity in the calculation in R [88]. Second, if the calculation may lead to a non-stationary process, despite the fact that the data has already been differenced once (marked by a double asterisk in Table 4.1, Table 4.2, Table 4.3, and Table 4.4). Calculation failures with the ML method were marked as '-' in Appendix A. The conditional sum of squares (CSS), (some refer to this method as OLS), replaced the ML method when calculation failure occurred. The use of CSS in parameter estimation can be seen with a red colour font in Appendix A. However, the calculation process can emerge to the non-stationary condition. This is a problem in using the ARIMA model as it requires human supervision on the calculation.

Different methods and configurations on the testing phase in the outlier module with the monthly data batch is plotted using box-and-whisker in Figure 4.20. Monthly data comprises three sequences, namely the first, second, and third months. The y-axis describes the percentage accuracy of all methods on the experiment, while the x-axis lists all tested scenarios (scenario1, scenario2, etc.). Since there are three sequences (or months) there were two results for each scenario, namely the accuracy of the scenario on the second and the

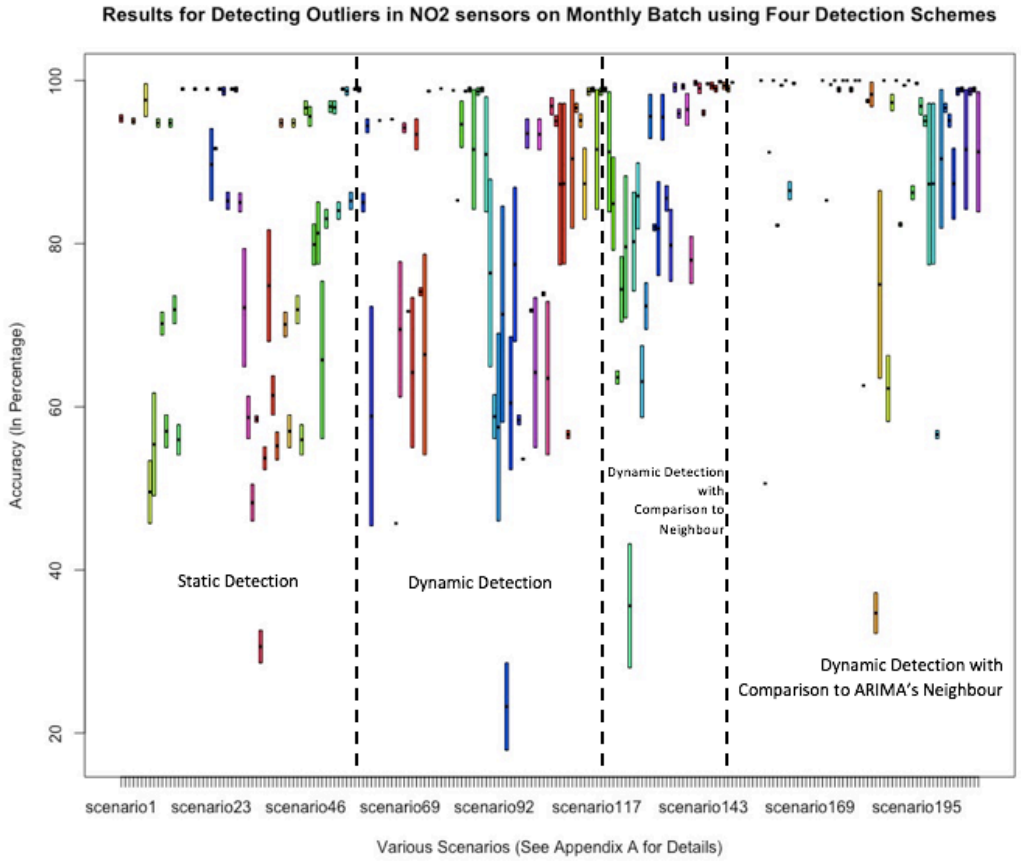
third month. For a complete list of scenarios, refer to Appendix A. The figures are divided into four proposed detection schemes.

The accuracy of the SD and DD schemes in the NO_2 sensors vary from 0% to 100% in Figure 4.20a. DDCN was slightly better than the previous two methods mentioned, while leveraging neighbours the ARIMA model did the most accurate prediction. The accuracy was expected to increase with the extension of the prediction range by doubling the standard deviation in DDCAN, particularly if the estimation parameter function was ML. Interestingly, doubling the standard deviation did not necessarily increase the prediction accuracy when CSS was used to determine parameters.

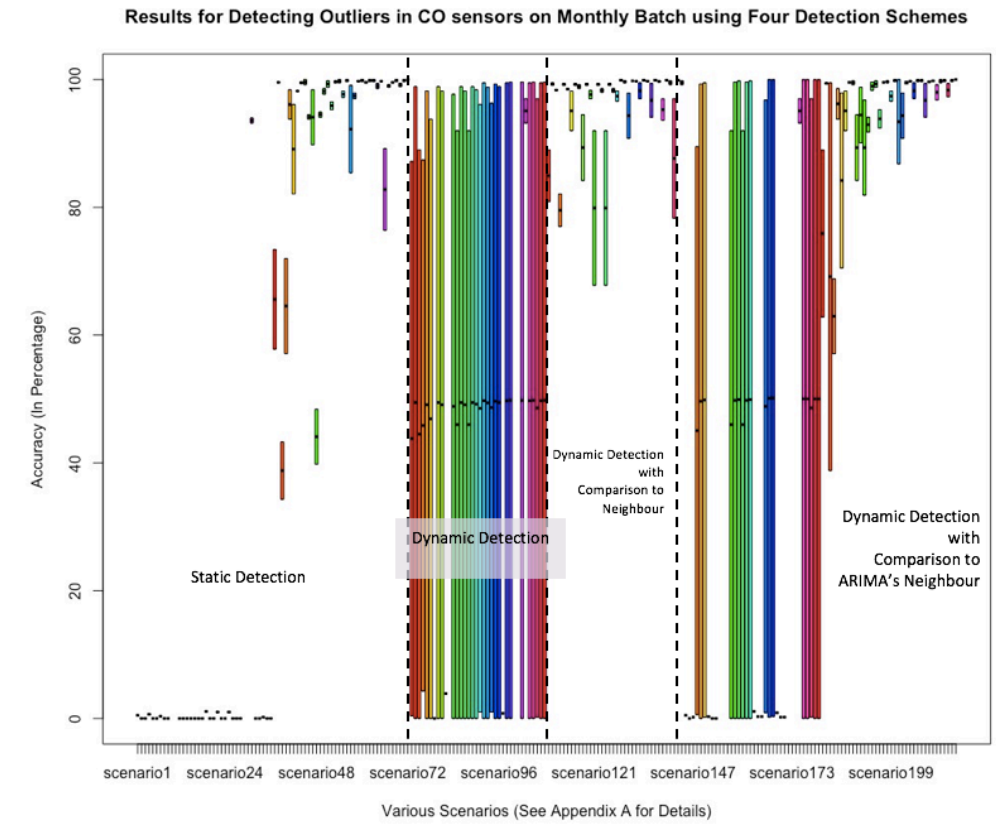
Figure 4.20b shows the CO sensors evaluation by various outlier schemes and configurations. Any ARIMA model with two standard deviations prediction range performed quite accurately (more than 90%) in the SD scheme using CSS. The DD scheme predicted well on the third month, but very poor on the second month. The DDCN scheme works slightly better than the previous two schemes. Overall, the DDCAN scheme can distinguish outliers better than the other three schemes, particularly using CSS.

Figure 4.20c describes the performance of outlier detection schemes for temperature sensors. The SD has less than 70% accuracy, DD is slightly better than SD but its performance varied a lot. The DDCAN scheme has less variance in accuracy. So, it is better than the previously mentioned methods. The best option for detecting outliers in the temperature sensor was the DDCN scheme. The accuracy of this scheme was 80% at worst. Seasonal ARIMA models using CSS or non-seasonal ARIMA models with ML had the most accurate prediction on this method.

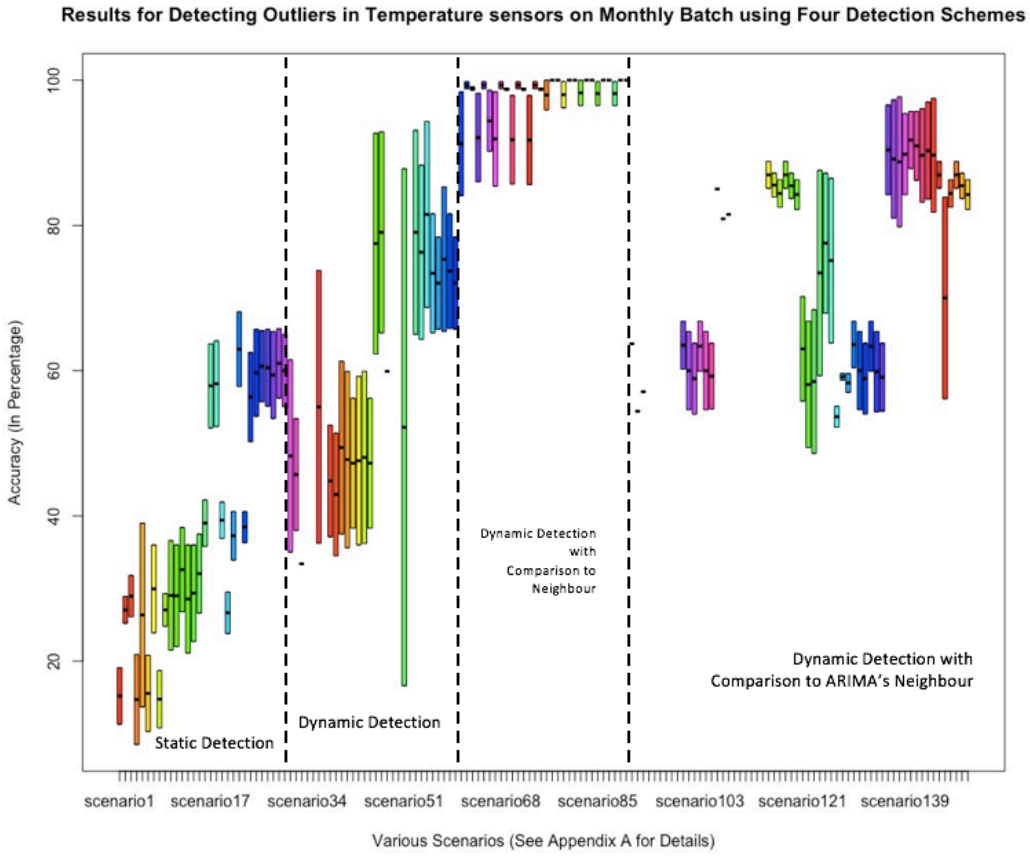
Figure 4.20d depicts the accuracy of detection schemes in detecting outliers on humidity sensors. The accuracy of detecting outliers in humidity sensors was similar to the temperature sensors. The SD method performed very poorly, while DDCN had the most accurate prediction.



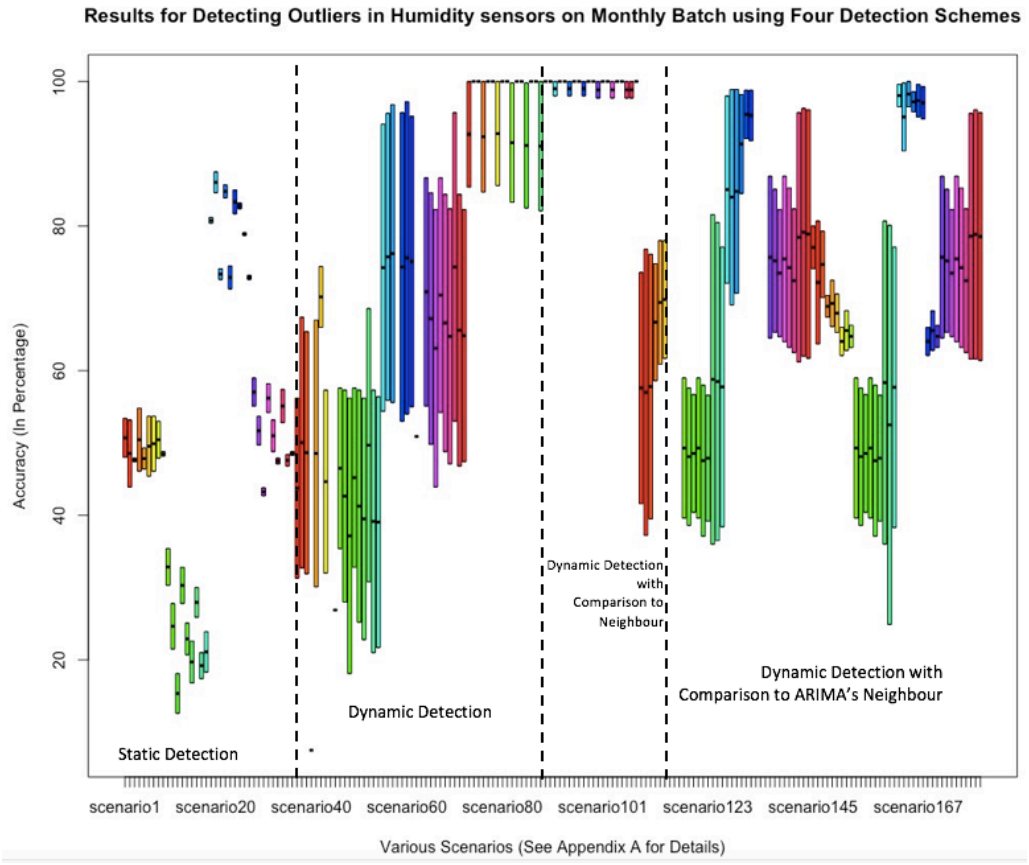
(a)


Results for Detecting Outliers in CO sensors on Monthly Batch using Four Detection Schemes
Accuracy (In Percentage)
0
20
40
60
80
100
Static Detection
Dynamic Detection
Dynamic Detection with Comparison to Neighbour
Dynamic Detection with Comparison to ARIMA's Neighbour
scenario1
scenario24
scenario48
scenario72
scenario96
scenario121
scenario147
scenario173
scenario199
Various Scenarios (See Appendix A for Details)

(b)


Results for Detecting Outliers in Temperature sensors on Monthly Batch using Four Detection Schemes
Accuracy (In Percentage)
20
40
60
80
100
Static Detection
Dynamic Detection
Dynamic Detection with Comparison to Neighbour
Dynamic Detection with Comparison to ARIMA's Neighbour
scenario1
scenario17
scenario34
scenario51
scenario68
scenario85
scenario103
scenario121
scenario139
Various Scenarios (See Appendix A for Details)

(c)



(d)

Figure 4.20 The accuracy results from various monthly detection schemes. (a) NO_2. (b) CO. (c) temperature. (d) humidity.

The box-and-whisker plot shows the performance of the four detection methods using the weekly data sequence in Figure 4.21 below. The vertical axis describes the accuracy percentage and the horizontal axis denotes scenarios in the testing phase. Since there were 12 batches (or weeks) there were eleven results for each scenario, starting from the second until the twelfth week. (Refer to Appendix A for a complete list of scenarios).

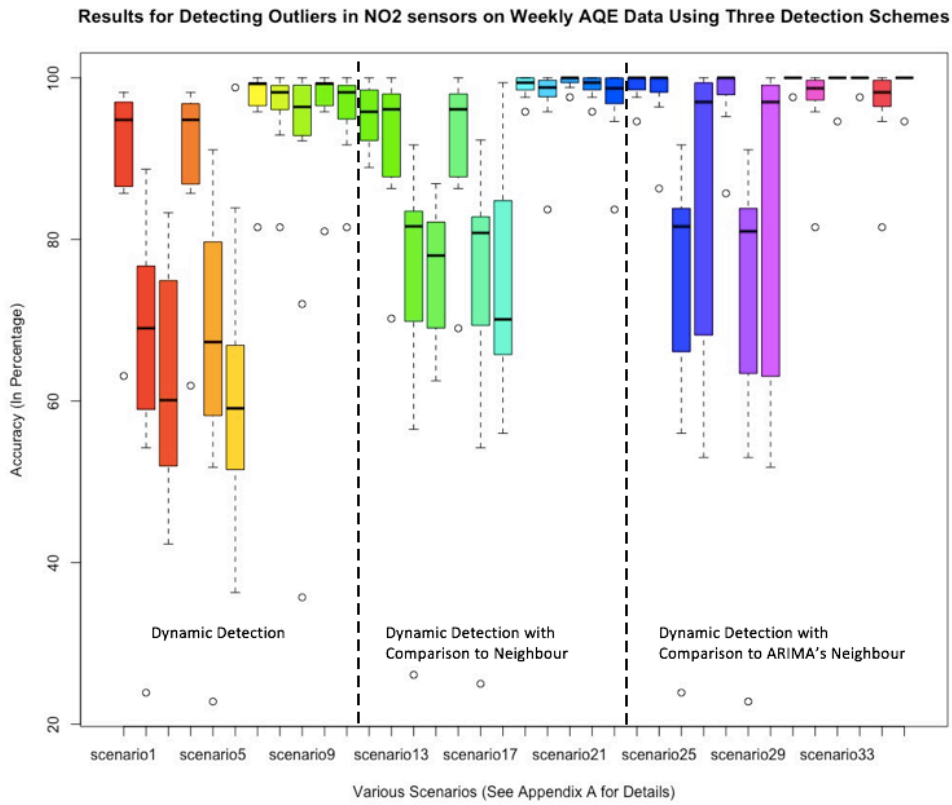
The SD was not considered in the weekly batch as it previously had poor performance in the monthly batch. Non-seasonal ARIMA estimators were also excluded on the testing phase since it cannot converge in the parameter calculation. The accuracy performance of DD was below 80% in predicting weekly NO_2 sensors, particularly using one standard deviation

range, as shown in Figure 4.21a. DD gained a better result of roughly 90% with two standard deviations prediction range. DDCN was better than the DD scheme. The best accuracy was the DDCAN scheme. The accuracy of this scheme had an improvement compared to DDCN. Generally, both schemes' checking the neighbour first before marking an outlier performed better at around 90% accuracy for NO_2 sensors.

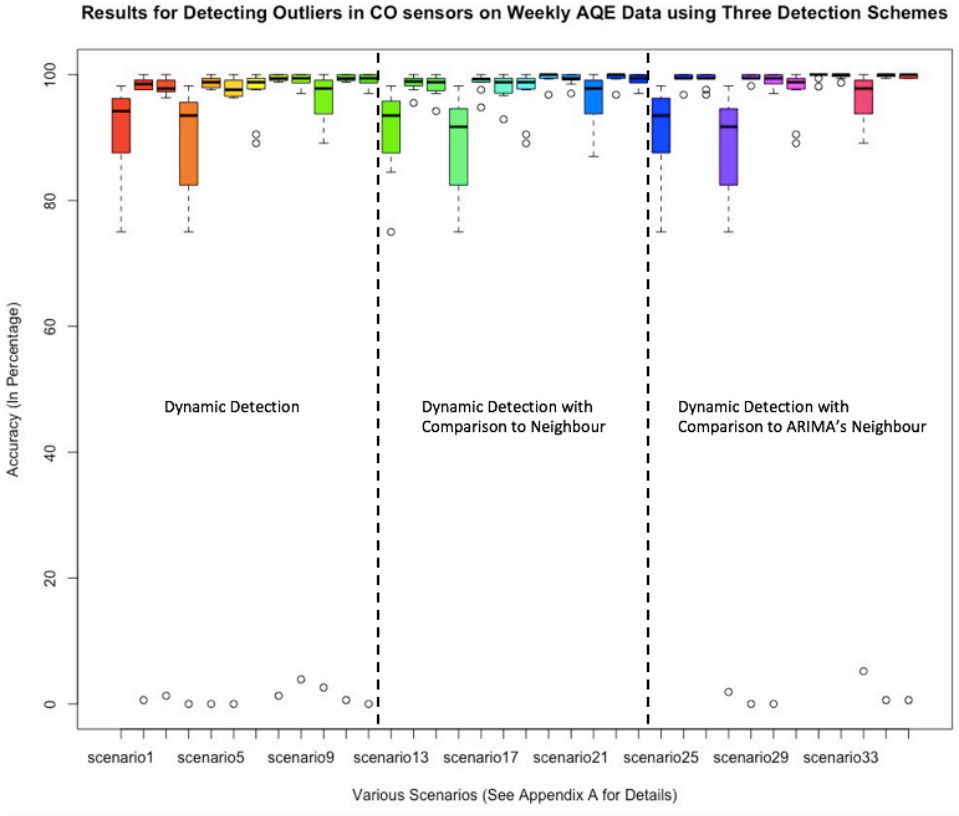
Figure 4.21b shows the performance of the three proposed detection schemes in detecting outliers on CO sensors with various configurations. Surprisingly, all scenarios had more than 80% accuracy in any of the batches. The performance of schemes checking neighbours before marking a suspected value outperform the DD scheme on using its own ARIMA prediction to mark outliers.

Figure 4.21c indicates the accuracy of temperature sensors in detecting outliers. Regarding accuracy, the DD scheme has poor performance compared to the other two schemes which are checking their neighbours before marking. Contrary to previous findings in the NO_2 and CO sensors detection schemes, DDCN outperformed DDCAN. DDCN was able to identify that there was no outlier in the batches with almost 100% accuracy.

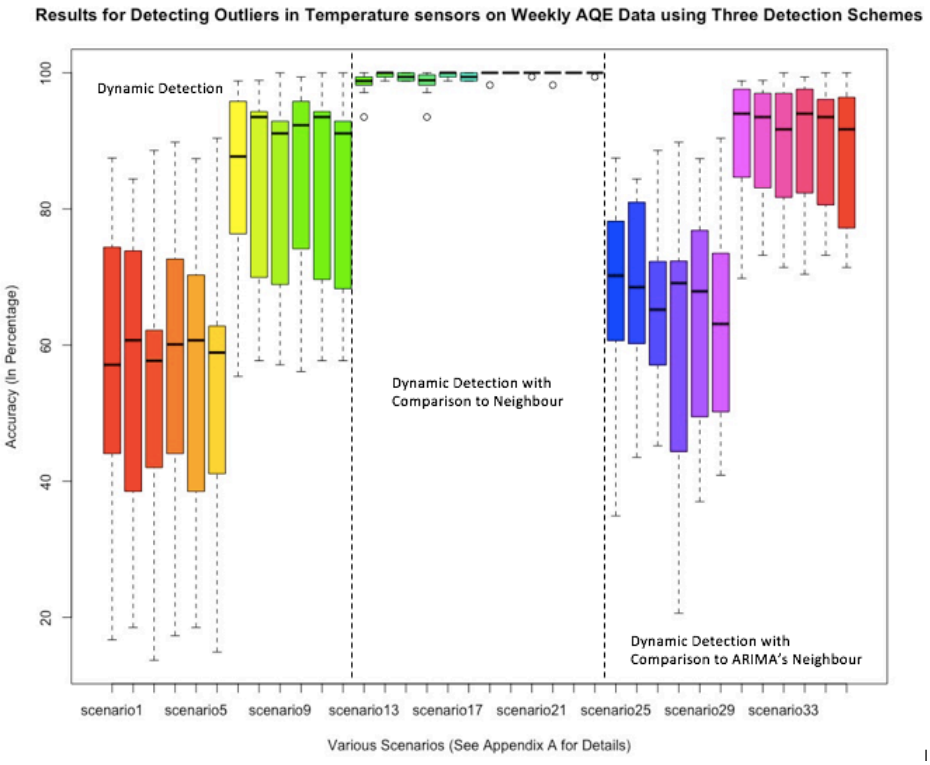
Figure 4.21d presents the accuracy percentage of the three proposed schemes on humidity sensors. The accuracy of DDCN was 100% in almost all batches. The scheme has the best accuracy to be used in the outlier module.


Results for Detecting Outliers in NO2 sensors on Weekly AQE Data Using Three Detection Schemes
Accuracy (In Percentage)
20
40
60
80
100
Dynamic Detection
Dynamic Detection with Comparison to Neighbour
Dynamic Detection with Comparison to ARIMA's Neighbour
scenario1
scenario5
scenario9
scenario13
scenario17
scenario21
scenario25
scenario29
scenario33
Various Scenarios (See Appendix A for Details)

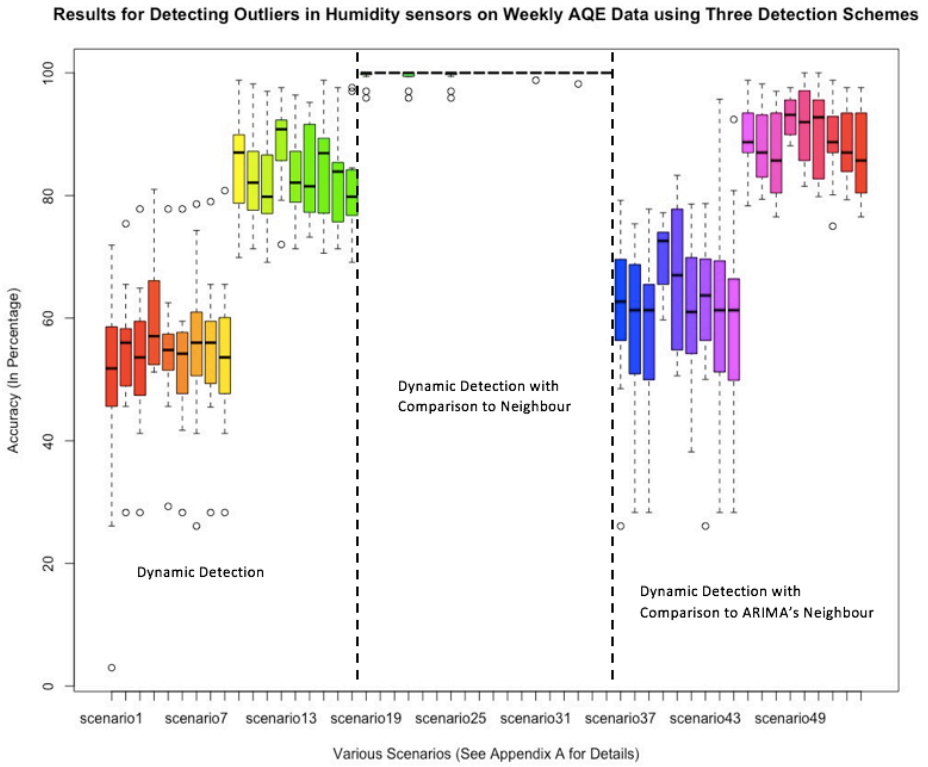
(a)


Results for Detecting Outliers in CO sensors on Weekly AQE Data using Three Detection Schemes
Accuracy (In Percentage)
0
20
40
60
80
100
Dynamic Detection
Dynamic Detection with Comparison to Neighbour
Dynamic Detection with Comparison to ARIMA's Neighbour
scenario1
scenario5
scenario9
scenario13
scenario17
scenario21
scenario25
scenario29
scenario33
Various Scenarios (See Appendix A for Details)

(b)



(c)



(d)

Figure 4.21 The accuracy results from various weekly detection schemes. (a) NO_2. (b) CO. (c) temperature. (d) humidity.

4.7 Summary

Two data sources obtained from AQEs and ECan were used in the study. Each AQE has four sensors to measure temperature, humidity, carbon monoxide, and nitrogen dioxide. Three AQEs were employed side by side with the ECan analysis instrument on the site. ECan and AQEs data were assessed statistically to obtain statistical information such as difference between sensor types, difference between the three AQE sensors, and readings agreement.

Past and current ECan data were used in the study. Past ECan was used to train the ARIMA models for various detection schemes in the outlier module during the training period and current ECan was used as a reference against the AQE data in order to assess the accuracy of the four proposed detection schemes in the testing phase. Several ARIMA models can be derived from the past two years' data (2014 and 2015) obtained from past ECan. The derivation of such estimators uses the Box-Jenkins approach. Seasonal and non-seasonal ARIMA models were explored in the training period. As a result, there are three ARIMA estimators each for temperature and CO sensors, and another four models each for humidity and NO_2 sensors. The estimators are then being employed to the detection schemes in the outlier module.

The four proposed detection schemes are: Static Detection (SD), Dynamic Detection (DD), Dynamic Detection with Comparison to Neighbours (DDCN), and Dynamic Detection with Comparison to ARIMA's Neighbours (DDCAN). The accuracy performance for each scheme is varied and depends on the dataset. The use of ARIMA in the detection schemes requires human supervision.

The sensors' readings agreement among adjacent nodes can help to determine an appropriate scheme to be applied in the outlier module. DDCN and DDCAN have the best accuracy in comparison to SD and DD. Particularly, DDCN can be used to detect outliers when there is a good reading agreement between sensor nodes, while DDCAN is employed if the sensor nodes tend to vary a lot in the readings. Parameter estimation using CSS is preferable over ML with the availability of small to medium size data. Non-seasonal ARIMA models are generally better than seasonal ARIMA models for detection schemes in the outlier module.

Chapter 5: Adjustment Module

This chapter describes the adjustment module for calibrating the AQE output in a supervised manner in two scenarios: with, and without the outlier module. Three methods are explored for the adjustment module, which will be explained further in Section 5.1. How the three methods are evaluated will be explained in Section 5.2. The training phase will be discussed in the Section 5.3. The testing phase with and without the outlier module inserted before the adjustment module, is discussed along with the findings in Section 5.4.

5.1 Proposed Methods

The three different methods employed in this research are Linear Regression (LR), Multi-Linear Regression (MLR), and Artificial Neural Network (ANN).

5.1.1 Linear Regression and Multi-Linear Regression

In LR, the linear model estimates independent variables into a dependent variable. In this research, each sensor on an AQE (x_t) is linearly fitted to its corresponding ECan data point (y_t). The three sensors, which are the CO or NO_2, temperature, and humidity, are being treated as the independent variables on each AQE.

In MLR, more independent variables from the other two AQEs are considered to be fitted to a single dependent variable from ECan. Three AQEs with their total of nine sensors (three sensors for each AQE) are fitted to their corresponding ECan dependent variables (CO or NO_2).

Rao and Toutenburg [89] explained the linear model concept used by LR in 1995. Suppose *Y* is a dependent variable that is related to *k* independent variables $X_1, \ldots, X_k$ and a random error *e* from a function *f*, the relationship of these variables can be defined as:

$$Y = f(X_1, \ldots, X_k) + e \qquad \textit{Equation 5.1}$$

If *f* is assumed to be a linear function, then a linear regression model of *Y* can be described as:

$$Y = X_1\beta_1 + \cdots + X_t\beta_k + e \qquad \textit{Equation 5.2}$$

Assuming there are *T* sets of observations on *Y* with its dependant variables $X_1, \ldots, X_k$, then the relationship between these variables in the matrix notation is:

$$(y, X) = \begin{pmatrix} y_1 & x_{11} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ y_t & x_{1t} & \cdots & x_{kt} \end{pmatrix} = \left(y, x_{(1)}, \ldots, x_{(k)}\right) = \begin{pmatrix} y_1, x_1' \\ \vdots \\ y_t, x_t' \end{pmatrix} \qquad \textit{Equation 5.3}$$

Thus, a general model of *t* observation for LR is

$$y = X\beta + e \qquad \textit{Equation 5.4}$$

with

$$X = \begin{pmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{1t} & \cdots & x_{kt} \end{pmatrix} \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}.$$

Ordinary Least Squares (OLS) is used in estimating β [89]. The aim of OLS is to estimate β by minimizing an error function *M* which is defined as:

$$\sum_{t=1}^{N} M(e_t) = \sum_{t=1}^{T} M(y_t - \beta x_t') \qquad \textit{Equation 5.5}$$

Suppose *B* contains the solution of vectors β. The goal is to obtain a vector b' = (b_1, …, b_k) from *B* which has the minimum of the sum of squared residuals:

$$S(\beta) = \sum_{t=1}^{T} e_t^2 = e'e = (y - X\beta)'(y - X\beta) \qquad \textit{Equation 5.6}$$

Solving *b*, there are two possible equations based on the number of unique solutions, according to Rao and Toutenburg. If a unique solution is available, *b* is defined as:

$$b = (X'X)^{-1}X'y \qquad \textit{Equation 5.7}$$

If there are several solutions to the regression, *b* is defined as:

$$b = (X'X)^{-1}X'y + (I - (X'X)^{-1}X'X)w \qquad \textit{Equation 5.8}$$

where:

$(X'X)^{-1}$is a generalized inverse of X'X, *w* is an arbitrary vector with n dimension, and *I* is an identity matrix.

5.1.2 Artificial Neural Network (ANN)

Three ANN methods are trained and tested in the adjustment module. These methods are selected because they had been implemented in the studies mentioned in section 2.3. The three ANN methods are briefly discussed below.

- *Single Hidden Layer Neural Network*

This is the simplest ANN architecture, since it adds only one layer between the input and the output layer [74]. A feed-forward network topology was implemented and illustrated in Figure 5.1. A vertex is denoted by a circle, and has a transfer function, while an edge indicates the information flow between the two vertices and has an assigned value called its weight. The network aims to find appropriate weights for all edges with the minimum loss function. Three AQE each with three sensors (CO or NO_2, temperature, and humidity sensors) feed the input layer. The data are split into two parts. The first part is to train the network, and the other part is to test the network performance.

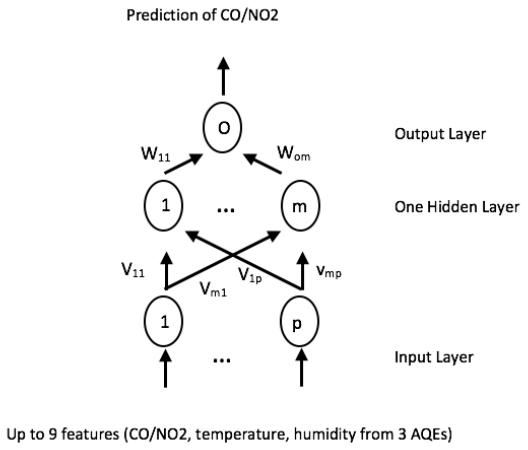


Figure 5.1 Architecture of single hidden layer neural network

- *Gradient Boosting*

Ridgeway's [90] implementation of gradient boosting is also used in this research. Gradient boosting initializes trees (or base learners) where input maps onto output. The learners keep updating their mapping during the training by minimizing loss function on each *M* iteration. Having several base learners can decrease a particular loss function. Instead of having a hidden layer, gradient boosting utilises trees to determine a response or output. The method is explained in great detail by Ridgeway [91]. Gradient boosting assumes that the data has a Gaussian distribution. The data is split into two parts, training and testing. The training part is further split into training and calibration data. Splitting of the training data is known as *bagging* and the percentage between the training and calibration data is determined by the bagging fraction. The training data is used to train a network, while the calibration data checks the network's configuration and modifies the network accordingly, based on a function. In this research, three functions were being tested to calibrate the network: out-of-bag (OOB), test set (test), and cross-validation (CV) [92]. The second part is to test the network using a different set of data apart from the training and calibration data. The output of the testing part is the prediction of the network. In this research, the prediction

was then compared to the ECan reference using three evaluation methods. This data training separation does not apply to Single Hidden Layer or Multi-Layer Perceptron networks.

- *Multi-Layer Perceptron (MLP)*

By adding more layers to the structure, an MLP could have more than one layer in the middle between the input (explanatory variables) and output (response variable). The MLP architecture is a feed-forward network with backpropagation. Figure 5.2 illustrates a typical MLP architecture, with no calibration phase. The implementation of MLP was applied using the Stuttgart Neural Network Simulator (SNNS) [93]. Again, three AQEs, each with three sensors (CO or NO_2, temperature, and humidity sensors), feed the input layer.

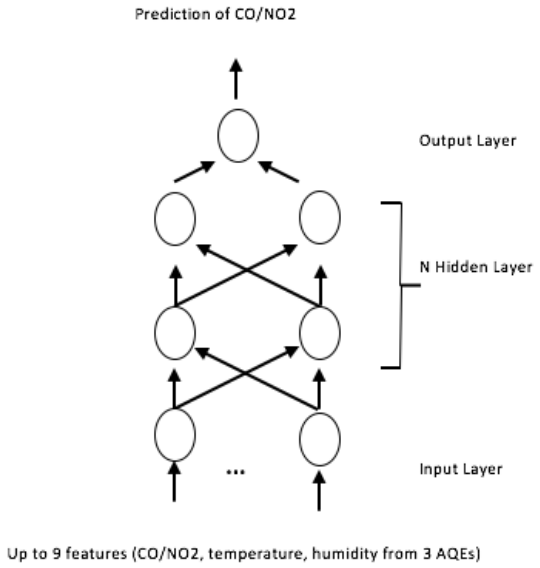


Figure 5.2 A multi-layer perceptron topology

5.2 Evaluation Methods

The adjustment module fits the AQE data into the corresponding ECan data. Three evaluation methods were used to assess the fitting performance of the adjustment module. In terms of evaluation methods, Willmott [83, 94] discouraged relying on just one method for

evaluation, while Comrie [83, 94] argued that the coefficient of determination, despite being a common method, may not be suitable for evaluating air quality prediction. Therefore, the coefficient of determination (*R-square*), root mean squared error (RMSE), and index of agreement (*d*) were used as evaluation measures.

Evaluation methods compare the output of a method's prediction against the reference, within a certain time period, in both the training and testing phase. The evaluation methods measure how well a method and a network fit the reference in those two stages. They are useful to determine the best fitting performance of the method and the network. The testing phase was evaluated using the three evaluation methods explained below, while the training phase was assessed solely with the *R-square* method because we were only interested in the performance of the testing phase.

5.2.1 Coefficient of Determination (*R-Square*)

Normally, the result of *R-square* ranges from zero to one where zero correlates to a very poor performance and one correlates to the ability of the method's response to suit the target reference perfectly. It is possible that *R-square* can give a result greater than 1, but it is unlikely to happen. The calculation of *R-square* is:

$$r^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} \qquad \textit{Equation 5.9}$$

where $\hat{y}$= the predicted value, y_i= the observed value, $\bar{y}$= mean of the observed value.

5.2.2 Root Mean Square Error (RMSE)

While *R-square* indicates the relative goodness between prediction and reference, RMSE is an absolute measure to understand the standard deviation of the unknown factors in variance.

The RMSE is defined as:

$$rmse = \sqrt{\frac{\sum(\hat{y}_i - y_i)^2}{N}} \qquad \textit{Equation 5.10}$$

where $\hat{y}$= the predicted value, y_i= the observed value, N = the number of data

5.2.3 Index of Agreement (*d*)

The agreement index shows the relationship between the prediction and the reference. Its output ranges from zero to one, which is similar to *R-square*. Legates and McCabe [95] argued that *R-square* does not count outliers, additive, and proportional differences which may exist in prediction and reference. So, the index of agreement was used as an alternative measurement.

The index of agreement is defined as:

$$d = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum[|\hat{y}_i - \bar{y}| + |y_i - \bar{y}|]^2} \qquad \textit{Equation 5.11}$$

where $\hat{y}$ = the predicted value, y_i = the observed value, $\bar{y}$ = mean of the observed value.

5.3 Training Phase

The training of LR, MLR, and ANN are discussed in this section. The result will be used and evaluated for the testing phase in section 5.4.

5.3.1 Linear Regression and Multi Linear Regression

Calculations and simulations were conducted based on these following scenarios:

(i). Target variable of LR testing simulations involves a single CO or NO_2 sensor. The AQE data was split into training percentages of 50%, 75%, and 90% from the dataset for each AQEs. There were 18 simulations in total for fitting the AQE's CO or NO_2 sensor into their corresponding ECan gas. Take AQE1 as an example, its CO sensor was fitted to the ECan using 50% data from AQE1. Another example is that the AQE1's CO sensor was fitted to the ECan using 75% of its data.

(ii). MLR scenarios fitted CO or NO_2 data from ECan using three variables from the corresponding CO or NO_2 sensors on AQEs. In another scenario, MLR tested the additional 6 variables taken from temperature and humidity sensors on AQEs to add up to 9 variables to be considered as the explanatory variables. Both the two main scenarios ran the simulation with various data training percentages of 50%, 75%, and 90%. In total, there were 12 simulations of CO and NO_2.

5.3.2 Artificial Neural Network (ANN)

The ANN's supervised learning method was employed in this research. The CO and NO_2 sensors on the three AQEs were the inputs of the network along with their temperature and humidity sensors. Combining all the sensors' output, there were 9 features or independent variables to the CO or NO_2 networks. The network's target response was CO or NO_2 obtained from ECan.

ANN tends to have a better result for big data as it allows the technique to fully capture the whole domain problem from the data [96]. Hence, larger datasets help the ANN in inferring a good generalization. Basheer and Hajmeer [96] argued that the use of ANN demands the data be partitioned into three subsets, specifically training, calibration, and testing. Training and calibration subsets are used by ANN to learn and improve its internal network. ANN determines its internal values by using the training subset and accelerates its calculated values accordingly based on the calibration subset. The testing subset is used by

the network to predict the output of a target (CO and NO_2 in this study). The prediction of the network then be used to compare against the true ECan reference. Having abundant data is important, so feature extraction and feature selection, which basically suggest to reduce the number of features, were not considered in the experiment because it can reduce the number of data supplied to the network. Meanwhile, the number of learning parameters, the number of iterations, the number of hidden units, and the type of activation function in the hidden and output layers were explored to obtain the best network configuration for fitting the ECan data. Detailed configurations from the three different architectures are discussed below.

- *Single Hidden Layer Network*

Running the topology, the calculation ran faster than the other architectures mentioned below. Data pre-processing, particularly data normalization, was applied to the CO and NO_2 sensors on AQEs. Data normalization is useful to block overriding of larger numbers to smaller ones in the learning process [96]. Another use of normalization is that it can stop premature saturation of hidden nodes. The data is normalized using the following formula [50]:

$$x_{norm} = 2 * \left(\frac{(x - x_{min})}{(x_{max} - x_{min})}\right) - 1 \qquad \textit{Equation 5.12}$$

Later, the data can be returned to the original units with the following formula:

$$x = \left(\frac{(x_{norm} + 1.0) * (x_{max} - x_{min})}{2}\right) + x_{min} \qquad \textit{Equation 5.13}$$

where x is the input series, x_{min} is the minimum value on the series, x_{max} is the maximum value on the series.

More than 19,000 scenarios for each NO_2 and CO sensor were conducted in the experiment. The scenarios involved a combination of different factors. The following factors were considered to be incorporated into the simulations without using the outlier module:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), or 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The number of nodes in a single hidden layer: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 nodes.

(iii). The activation functions in the output layer: tangent hyperbolic and logistic function.

(iv). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(v). The training percentage: 50%, 75%, 90%.

(vi). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

(vii). The type of input: normal input, and normalized input.

With the outlier module, the following factors, derived from Section 4.6, are considered in the simulation:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), or 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The number of nodes in a single hidden layer: 1, 2, 3, 4, 5, 6, 7, 8, 9 nodes.

(iii). The activation functions in the output layer: tangent hyperbolic and logistic function.

(iv). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(v). The training percentage: 50%, 75%, 90%.

(vi). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

(vii). ARIMA estimators: (0,1,5), (1,1,1), (1,1,1) x (12,1,12), (1,1,1) x (24,1,24), (0,1,5) x (12,1,12), (1,1,1) x (12,0,12).

(viii). Decision schemes: DDCN, DDCAN.

(ix). Parameter estimation: CSS, ML.

(x). Batches: weekly, monthly.

- *Gradient Boosting*

More than 5,000 scenarios for each CO and NO_2 sensor were assessed in the gradient boosting network. The following factors were combined in the simulations without the presence of the outlier module:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), and 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The types of iteration methods: CV, OOB, and test.

(iii). The fraction of bag: 50% and 75%.

(iv). The fold number of cross-validation: 5 and 10.

(v). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(vi). The training percentage: 50%, 75%, 90%.

(vii). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

Meanwhile, when the outlier module was present, the factors which are obtained from the evaluation of ARIMA estimators and decision schemes in Section 4.6 are:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), and 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The types of iteration methods: CV, OOB, and test.

(iii). The fraction of bag: 50% and 75%.

(iv). The fold number of cross-validation: 5 and 10.

(v). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(vi). The training percentage: 50%, 75%, 90%.

(vii). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

(viii). ARIMA estimators: (0,1,5), (1,1,1), (1,1,1) x (12,1,12), (1,1,1) x (24,1,24), (0,1,5) x (12,1,12), (1,1,1) x (12,0,12).

(ix). Decision schemes: DDCN, DDCAN.

(x). Parameter estimation: CSS, ML.

(xi). Batches: weekly, monthly.

- *Multi-Layer Perceptron (MLP)*

More than 7,000 and 3,000 scenarios of a respective NO_2 and CO sensor were tested in the MLP network. The following factors were combined on the simulations without the outlier module:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), or 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The activation function for hidden and output layers: tangent hyperbolic and logistic.

(iii). The training percentage: 50%, 75%, 90%.

(iv). The number of hidden layer (and nodes): 2 (3,3), 2 (9,9), 3 (3,3,3), 3 (3,3,1), 3 (3,5,3), 3 (9,9,1), 3 (9,9,9), 3 (9,5,9).

(v). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(vi). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

Evaluating the ARIMA estimators and decision schemes from the previous chapter, the simulation with the outlier module uses the following combinations:

(i). The number of explanatory variables: 3 (CO or NO_2 sensor from three AQEs), or 9 (three CO or NO_2, three temperatures, three humidity sensors from three AQEs).

(ii). The activation function for hidden and output layers: tangent, hyperbolic and logistic.

(iii). The training percentage: 50%, 75%, 90%.

(iv). The number of hidden layers (and nodes): 3 (3,6, 9), 3 (9, 6, 3).

(v). The learning parameter: 0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005.

(vi). The number of iterations: 1, 10, 100, 500, 1000, 5000, 10000, 50000.

(vii). ARIMA estimators: (0,1,5), (1,1,1), (1,1,1) x (12,1,12), (1,1,1) x (24,1,24), (0,1,5) x (12,1,12), (1,1,1) x (12,0,12).

(viii). Decision schemes: DDCN, DDCAN.

(ix). Parameter estimation: CSS, ML.

(x). Batches: weekly, monthly.

5.4 Testing Phase

The NO_2 or CO fitting involving solely one feature of an AQE sensor had failed to fit the reference as its R-square value is as low as 3% using LR. The fitting slightly gets better when all three AQE sensors of a corresponding gas were added to the explanatory variables using MLR. The NO_2 fitting rate increased when the temperature and humidity sensors were included. The AQEs' CO readings tend to agree with each other, as discussed in Section 4.5, while the AQEs' NO_2 readings tend to vary a lot. The reading variance may contribute to the fitting performance in the adjustment module. Generally, the ANN method with redundant nodes is better than both LR and MLR. The CO fitting using ANN without the outlier module is better than the NO_2 fitting. Meanwhile, the output of the adjustment module using ANN for both gases is better when the outlier module present. However, it is interesting to note that the number of the dataset was reduced by the outlier module before passing to the adjustment module. The result for each tested method is discussed in detail below.

5.4.1 Without Outlier Module

- *Linear Regression and Multi Linear Regression*

LR and MLR cannot be used to calibrate AQE for both NO_2 and CO gases concentration, because its prediction performance was far from the ECan reference. Table 5.1 selects the best performance result on NO_2 and CO prediction in the training and testing phases. The *R-square* of NO_2 and CO were a respective of 42% (0.423) and 33% (0.333) in the training phase

with 9 independent variables. The *R-square* for NO_2 was 3.7% (0.037) and CO was 2.9% (0.029) in the testing stage using the same training stage configuration regardless of the percentage of data training. The number of data did not impact on the LR and MLR performance.

INPUT DATA	OUTPUT DATA	TRAINING PERCENTAGE	r2 TRAINING	r2 TESTING	RMSE	d
AQE2	NO_2	0.5	0.065	0.038	11.347	0.813
AQE2	NO_2	0.75	0.065	0.038	11.347	0.813
AQE2	NO_2	0.9	0.065	0.038	11.347	0.813
AQE1, AQE2, AQE3, 3 Temp, 3 Hum	NO_2	0.5	0.423	0.037	15.106	0.75
AQE1, AQE2, AQE3, 3 Temp, 3 Hum	NO_2	0.75	0.423	0.037	15.106	0.75
AQE1, AQE2, AQE3, 3 Temp, 3 Hum	NO_2	0.9	0.423	0.037	15.106	0.75
AQE1	CO	0.5	0.125	0.031	22.002	0.625
AQE1	CO	0.75	0.125	0.031	22.002	0.625
AQE1	CO	0.9	0.125	0.031	22.002	0.625
AQE1, AQE2, AQE3	CO	0.5	0.333	0.029	22.018	0.625
AQE1, AQE2, AQE3	CO	0.75	0.333	0.029	22.018	0.625
AQE1, AQE2, AQE3	CO	0.9	0.333	0.029	22.018	0.625

Table 5.1 Result of linear regression and multi linear regression

- *Artificial Neural Network*

Tens of thousands of simulations described in Section 5.3 were conducted in order to find the right network configuration, particularly for finding the appropriate configuration for fitting NO_2. The best fitting result was achieved when 90% data was used for training. It appears that about 2,000 rows were not sufficient enough to train the ANN for fitting the NO_2 gas, especially since the calculation fully depends on 9 variables. Having nine independent variables as the inputs to the ANN was considered to be a small number of variables in the ANN context [96]. R-square of less than 40% on the training phase indicated that the network was never able to fit the training data fully. However, the fitting of CO readings using ANN is impressive. The ANN was able to fit the CO reference by 80%. The discussion of the three ANNs architecture is presented below.

- Single Hidden Layer (SHL)

Thousands of simulations were conducted. Table 5.2 presents only the compelling simulation results. It shows that the *R-square* values are larger than 13% and 74% for NO_2 and CO, respectively, in the testing phase. The NO_2 or CO fitting have a better performance when all three sensors on an AQE were considered. Fitting CO using the logistic function on the output layer worked better than the tangent hyperbolic function, while fitting NO_2 using the tangent hyperbolic function on the output layer gave the highest r-square in the testing phase. The fitting of CO using single hidden layer may consider several network configurations with nearly similar *R-square* values of 75% and a small RMSE of less than 0.2 on the data test set regardless the number of nodes in the hidden layer. The index of agreement (*d*) generally agreed with the *R-square* where the *d* value increased when the R-square increased for different tested parameters. The configuration for the NO_2 single layer network can use a small learning parameter of at least 0.001 and a minimum iteration of 100. It is interesting to note that a relatively high RMSE value of roughly 65 occurred when the NO_2 sensors were used; while a very small RMSE value of 0.5 occurred when the corresponding NO_2 sensors got normalized before being fed into the network. The high RMSE in the testing phase for the NO_2 sensors was too high in comparison to a difference between AQE1 and AQE2 (16.82) in Section 4.5.2.

Normalized input may encounter challenges in the process of converting back the data into its normal units. Although the normalized NO_2 inputs fit relatively better than the normal ones on the simulations, there is a possibility that the denormalized data may result on a poor fitting, particularly because converting the data requires a one-year dataset in order to fully

understand the seasonal variation. Incomplete understanding of the event may lead to incorrect minimum and maximum values during the observed period.

INDEPENDENT VARIABLES	TARGET VARIABLE	NUMBER OF NODES	OUTPUT FUNCTION	LEARNING PARAMETER	EPOCH	TRAINING PHASE r2	TESTING PHASE			TRAINING PERCENTAGE	REMARK
							r2	RMSE	d		
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	6	tanh	0.00005	1000	0	0.201	62.418	0.242	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	7	tanh	0.00005	50000	0.011	0.196	63.212	0.238	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	8	tanh	0.0005	10000	0.076	0.263	68.347	0.217	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	6	logistic	0.5	10	0.029	0.231	28.427	0.414	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	1	logistic	0.00005	100	0.016	0.16	0.507	0.451	0.9	Normalized AQEs
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	2	logistic	0.00005	5000	0.009	0.176	0.507	0.451	0.9	Normalized AQEs
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	7	logistic	0.01	10	0.014	0.156	0.507	0.451	0.9	Normalized AQEs
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	1	logistic	0.001	5000	0.372	0.75	0.185	0.79	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	1	logistic	0.0001	5000	0.373	0.751	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	1	logistic	0.0001	50000	0.373	0.751	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	1	logistic	0.0005	100	0.373	0.751	0.185	0.789	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	2	logistic	0.001	100	0.372	0.75	0.184	0.79	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	2	logistic	0.001	50000	0.372	0.75	0.185	0.79	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	2	logistic	0.0001	100	0.372	0.752	0.184	0.791	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	2	logistic	0.0005	500	0.373	0.751	0.185	0.789	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	2	logistic	0.0005	5000	0.373	0.751	0.185	0.789	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	3	logistic	0.001	100	0.371	0.751	0.183	0.799	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	3	logistic	0.001	50000	0.372	0.75	0.185	0.79	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	3	logistic	0.0001	500	0.374	0.751	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	3	logistic	0.0005	100	0.373	0.751	0.185	0.789	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	3	logistic	0.00005	5000	0.374	0.755	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	4	logistic	0.0001	5000	0.373	0.751	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	4	logistic	0.00001	100	0.372	0.75	0.186	0.785	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	4	logistic	0.00001	500	0.374	0.752	0.185	0.787	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	5	logistic	0.001	1000	0.372	0.75	0.185	0.79	0.9	

AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	5	logistic	0.00001	1000	0.374	0.752	0.185	0.787	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	5	logistic	0.00001	50000	0.374	0.752	0.185	0.787	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	5	logistic	0.00005	100	0.372	0.75	0.185	0.79	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	6	logistic	0.0001	100	0.37	0.752	0.179	0.808	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	7	logistic	0.00005	500	0.374	0.752	0.185	0.788	0.9	
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	8	logistic	0.00001	100	0.372	0.752	0.186	0.786	0.9	

Table 5.2 Selected result of single hidden layer network

- Gradient Boosting

The simulation results on the testing phase are shown in the Table 5.3. The table shows *R-square* of more than 34% (NO_2) and 82% (CO) in the testing phase. The RMSE value was not as high as the difference between AQE1 and AQE2 (16.82) for NO_2. CO has a very small RMSE of less than 0.2. The *R-square* and *d value* concur each other. The simulation showed that cross-validation and out-of-bag iteration methods outperform out-of-sample. 50% bagging was enough to train the network, although there was some evidence that using 75% could perform well at fitting the test phase dataset. Fitting CO sensors may use a learning parameter at maximum of 0.1, while using 0.5 for NO_2 sensors. A small epoch can be applied when using the out-of-bag iteration method in fitting NO_2, while a big epoch is applied for the cross-validation technique. However, the same principle cannot be applied for fitting CO.

INDEPENDENT VARIABLES	TARGET VARIABLE	ITERATION METHOD	BAG %	CV	TRAINING PERCENTAGE	LEARNING PARAMETER	EPOCH	TRAINING PHASE r2	TESTING PHASE		
									r2	Rmse	d
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	cv	0.5	5	0.9	0.05	1	0.182	0.352	15.969	0.587
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	cv	0.5	5	0.9	0.05	1000	0.185	0.351	16.089	0.587
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	cv	0.5	5	0.9	0.05	10000	0.185	0.37	16.05	0.588
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	cv	0.5	5	0.9	0.05	50000	0.186	0.352	16.229	0.584
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.5	5	0.9	0.05	1	0.184	0.37	16.084	0.588
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.5	10	0.9	0.05	1	0.188	0.351	16.034	0.584
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.5	10	0.9	0.05	10	0.187	0.359	15.831	0.586
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.5	10	0.9	0.05	500	0.191	0.354	15.816	0.585
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.5	10	0.9	0.05	1000	0.179	0.353	15.93	0.587
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	NO_2	OOB	0.75	10	0.9	0.5	5000	0.188	0.402	16.78	0.588

AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	cv	0.5	10	0.9	0.05	10	0.383	0.83	0.177	0.843
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	cv	0.5	10	0.9	0.05	5000	0.387	0.832	0.185	0.832
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	cv	0.75	5	0.9	0.05	100	0.384	0.834	0.187	0.828
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.5	5	0.9	0.1	500	0.378	0.83	0.169	0.863
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.5	5	0.9	0.1	10000	0.38	0.83	0.168	0.866
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.5	5	0.9	0.05	5000	0.385	0.834	0.185	0.83
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.5	10	0.9	0.05	100	0.382	0.834	0.182	0.836
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.5	10	0.9	0.05	50000	0.382	0.831	0.182	0.838
AQE1,AQE2, AQE3, 3 Temp, 3 Humidity	CO	OOB	0.75	10	0.9	0.1	100	0.374	0.832	0.171	0.858

Table 5.3 Selected result of gradient boosting

- Multi-Layer Perceptron

Although MLP architecture has more flexibility in building the network, finding the right architecture for getting the best fit between ECan and the AQEs was a challenging task, as there would be a plethora of combinations. Table 5.4 presents the simulations with more than 30% and 60% *R-square* results in the testing phase. The optimum result was with three hidden layers using a logistic function on the hidden and output layers for NO_2 concentration, and the tangent hyperbolic function on all layers for CO. However, the simulation for NO_2 was not fully satisfactory as the RMSE values were higher than the statistical difference between AQEs (16.82). Fitting CO, on the other hand, was considered to give a fairly satisfactory result of more than 60% similarities to reference with RMSE of 0.2.

INDEPENDENT VARIABLES	TARGET VARIABLE	HIDDEN FUNCTION	OUTPUT FUNCTION	PERCENTAGE TRAINING	NUMBER OF NODES (HIDDEN LAYER)	LEARNING PARAMETER	EPOCH	TRAINING PHASE r2	TESTING PHASE		
									r2	RMSE	d
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	NO_2	Logistic	Logistic	0.9	9,5,9 (3)	0.1	1	0.09	0.35	28.43	0.41
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	NO_2	Logistic	Logistic	0.9	3,3,3 (3)	0.001	1	0.12	0.31	28.43	0.41
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	NO_2	Logistic	Logistic	0.9	9,9,9 (3)	0.00005	1000	0	0.33	28.43	0.41
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	NO_2	Logistic	Logistic	0.9	9,9,1 (3)	0.00001	1	0.08	0.35	28.88	0.41
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	CO	Tangent H	Tangent H	0.9	3,5,3 (3)	0.0001	50000	0.34	0.69	0.21	0.78
AQE1,AQE2,AQE3, 3 Temp, 3 Hum	CO	Tangent H	Tangent H	0.9	3,5,3 (3)	0.00005	50000	0.37	0.63	0.19	0.79

Table 5.4 Selected result of multi-layer perceptron

Considering all three ANN networks, it is interesting that simulations on different ANN architectures and two different running environments give some variability in the results,

particularly on gradient boosting and MLP structure. This might be partly due to the high RMSE rate on NO_2 simulations which brings high variance on the output, and partly because of the stochastic process in the estimation procedure which happens to be a statistical problem, according to Murata *et al* [97]. Murata *et al* explained that a stochastic process may occur when the data does not capture the whole domain problem. The estimation method only relied on the empirical distribution of the training set. As a result, the parameters, such as weights and thresholds, tended to change arbitrarily in the simulations.

From the discussion of the three ANN architectures above, fitting CO achieved 80% similarity compared to the ground truth using the gradient boosting method. Meanwhile, fitting NO_2 gave the best fitting of 40% to the ECan data using the same method. It seems that the reading agreement between CO sensors on AQEs has helped the ANN to calibrate its output easily. Meanwhile, a variation on NO_2 sensors has made it hard for ANN in the testing phase.

After evaluating the simulation results, the configuration with the highest *R-square* in the testing phase for NO_2 sensors is:

(i) Out-of-bag iteration method.
(ii) 0.75 bag fraction.
(iii) 10-fold cross-validation.
(iv) 90% data was used to train the network.
(v) Learning parameter of 0.5.
(vi) An epoch of 5,000.

The highest *R-square* in the testing data for predicting CO sensors has the following configuration:

(i) Out-of-bag iteration method.

(ii) Bag fraction of 0.5.

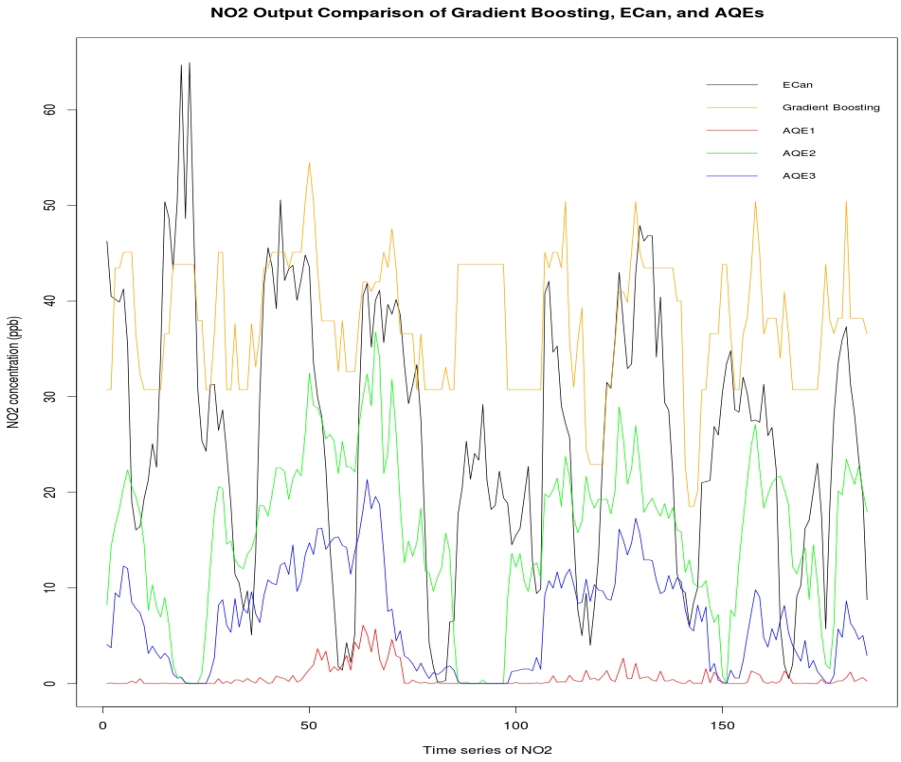
(iii) 5-fold cross-validation.

(iv) 90% data for training.

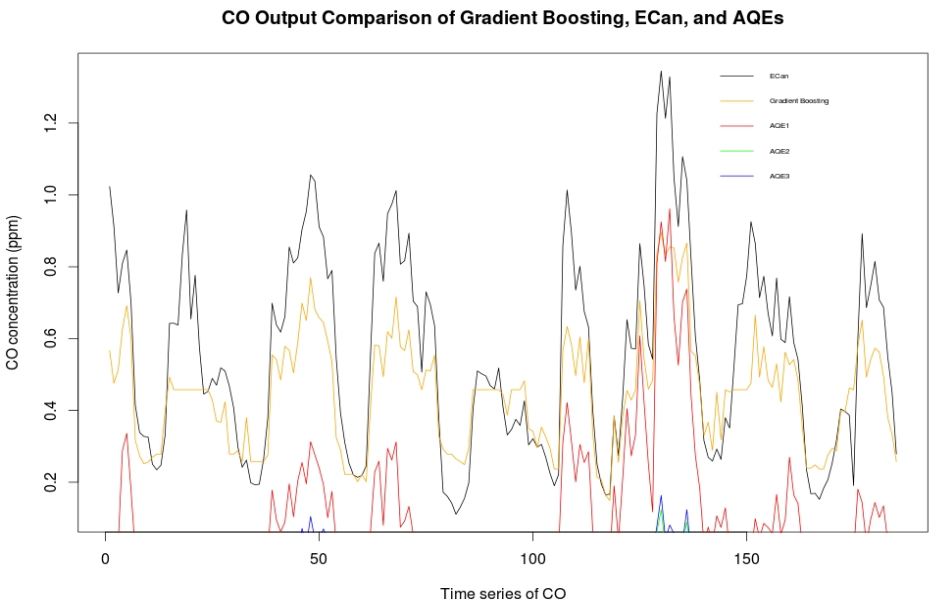
(v) Learning parameter of 0.05.

(vi) A repetition of 5,000.

Figure 5.3 plots the last 185 hours (10%) of AQE output prediction using the selected two configurations of gradient boosting in comparison with the corresponding AQE sensors and ECan. Figure 5.3a depicts the output of NO_2 and Figure 5.3b the CO output.



(a)



(b)

Figure 5.3 185-hour comparison of gradient boosting, reference, and AQEs on the concentration of:(a) nitrogen dioxide, and (b) carbon monoxide

5.4.2 With Outlier Module

Simulations with the presence of the outlier module were conducted. These simulations test the feasibility of using the outlier module before the adjustment module. We are interested to investigate whether the outlier module has contributed to the calibration of the adjustment module and whether the presence of the outlier module will benefit to the adjustment module. The simulations were performed with the use case scenarios described in Section 5.3. The performance result of the outlier module using seasonal and non-seasonal ARIMA estimators was obtained. The CSS was used to estimate the parameter in monthly batches and the ML in weekly batches. The use of the outlier module did enhance the calibration of the adjustment module for detecting CO and NO_2, particularly using the non-seasonal ARIMA estimators. The *R-square* of the adjustment module with the outlier module is higher than the *R-square* of solely using the adjustment module. The evaluation of both type of ARIMA estimators is discussed below.

- *Outlier Module with Seasonal ARIMA estimator*

The best performance of each neural network topology, as measured with *R-square* in the testing phase, is presented in Table 5.5, Table 5.6, and Table 5.7. The three tables show the simulation result with seasonal ARIMA estimators. The best *R-square* for the adjustment module without the outlier module in detecting CO was 0.8 and 0.4 for NO_2. Looking at the tables, the best R-square using seasonal ARIMA estimator can be reached at 0.68 for CO and 0.52 for NO_2, both use up 90% (0.9) data for the training phase. Note that there were about 540 rows (out of 2,006 rows) in the dataset passing to the adjustment module by the outlier module. There is an increase by 0.12 (12%) in calibrating NO_2, but a decrease in calibrating CO when the outlier module with seasonal ARIMA estimator was installed.

INDEPENDENT VARIABLES	TARGET	NUMBER OF NODES	OUTPUT FUNCTION	LEARNING PARAMETER	EPOCH	TRAINING PHASE R^2	TESTING PHASE			TRAINING PERCENTAGE	ARIMA ESTIMATOR, DECISION SCHEME, PARAM. ESTIMATOR	BATCHES
							R^2	RMSE	d			
3 AQEs, 3 Temp, 3 Humidity	CO	2	Tanh	0.01	100	0.218	0.673	0.865	0.439	0.9	(1,1,1) x (12,1,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	CO	4	Tanh	0.00005	500	0.311	0.674	1.491	0.209	0.9	(1,1,1) x (24,1,24). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	CO	10	Tanh	0.001	100	0.31	0.68	0.593	0.36	0.9	(1,1,1) x (12,0,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	4	tanh	0.00001	10000	0	0.488	55.878	0.228	0.9	(1,1,1) x (12,0,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	5	Tanh	0.00001	10000	0	0.488	55.878	0.228	0.9	(1,1,1) x (12,1,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	5	sigmoid	0.5	100	0.016	0.456	58.198	0.243	0.75	(1,1,1) x (24,1,24). DDCN. CSS	MONTHLY

Table 5.5 Selected result of single hidden layer with seasonal ARIMA estimator

INDEPENDENT VARIABLES	TARGET	NUMBER OF NODES (HIDDEN LAYER)	OUTPUT FUNCTION	LEARNING PARAMETER	EPOCH	TRAINING PHASE R^2	TESTING PHASE			TRAINING PERCENTAGE	ARIMA ESTIMATOR, DECISION SCHEME, PARAM. ESTIMATOR	BATCHES
							R^2	RMSE	d			
3 AQEs,	CO	3,6,9 (3)	Tanh	0.05	50000	0.038	0.487	0.292	0.529	0.5	(1,1,1) x (12,0,12). DDCAN. CSS	MONTHLY
3 AQEs	CO	9,6,3(3)	Tanh	0.5	10000	0.104	0.229	0.256	0.54	0.75	(1,1,1) x (12,0,12). DDCN.CSS	MONTHLY
3 AQEs	CO	9,6,3(3)	Tanh	0.5	5000	0.468	0.648	0.308	0.535	0.75	(1,1,1) x (12,0,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	3,6,9(3)	Logistic	0.00005	1	0.11	0.52	24.44	0.41	0.9	(1,1,1) x (12,0,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	9,6,3(3)	Logistic	0.00001	100	0.08	0.44	24.95	0.41	0.9	(0,1,5) x (12,1,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	9,6,3(3)	logistic	0.0001	1	0.1	0.5	24.33	0.41	0.9	(1,1,1) x (12,1,12). DDCN. CSS	MONTHLY

Table 5.6 Selected result of multi-layer perceptron with seasonal ARIMA estimator

INDEPENDENT VARIABLES	TARGET VARIABLE	ITERATION METHOD	BAG %	CV	TRAINING PERCENTAGE	LEARNING PARAMETER	TRAINING PHASE R^2	TESTING PHASE			EPOCH	ARIMA ESTIMATOR, DECISION SCHEME, PARAM. ESTIMATOR	BATCHES
								R^2	RMSE	d			
3 AQEs, 3 Temp, 3 Humidity	CO	Test	0.5	5	0.9	0.001	0.007	0.536	0.236	0.27	100	(1,1,1) x (12,0,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	CO	Test	0.5	5	0.75	0.005	0	0.55	0.275	0.4	100	(1,1,1) x (12,0,12). DDCN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	CO	Test	0.5	10	0.9	0.00001	0.012	0.504	0.236	0.27	1000	(1,1,1) x (12,0,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	Test	0.5	10	0.9	0.5	0.175	0.45	10.66	0.51	500	(0,1,5) x (12,1,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	Test	0.5	10	0.9	0.1	0.12	0.47	11.1	0.52	10000	(1,1,1) x (12,0,12). DDCAN. CSS	MONTHLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	Test	0.75	10	0.9	0.1	0.14	0.47	11.74	0.47	50000	(1,1,1) x (12,1,12). DDCN. CSS	MONTHLY

Table 5.7 Selected result of gradient boosting with seasonal ARIMA estimator

- *Outlier Module with Non-Seasonal ARIMA Estimator*

The best performance of the adjustment module where the outlier module was deployed with non-seasonal ARIMA estimators, measured with *R-square* in the testing phase, is presented in Table 5.8 and Table 5.9. The GBM network was not able to perform because of insufficient data, as the number of AQE readings had been reduced by the outlier module. An *R-square* of 0.97 for detecting CO and 0.65 for NO_2 using SHL can be inferred from Table 5.8. Better R-square of more than 0.9 for both gasses achieved using MLP in Table 5.9. Looking from the *R-square* values of the weekly batch, the prediction of the adjustment module with the outlier module activated has increased the fitting to the reference. However, it comes with a trade-off where the number of AQE readings had been removed to 90 (out of about 2000 rows) when the dataset comes to the adjustment module. The higher value of RMSE in NO_2 also indicates the prediction can greatly vary in the simulation. To employ the outlier module before the adjustment module, a mechanism is needed to keep the removal dataset so that the users will not perceive missing readings in the front end.

INDEPENDENT VARIABLES	TARGET	NUMBER OF NODES	OUTPUT FUNCTION	LEARNING PARAMETER	EPOCH	TRAINING PHASE R^2	TESTING PHASE			TRAINING PERCENTAGE	ARIMA ESTIMATOR, DECISION SCHEME, PARAM. ESTIMATOR	BATCHES
							R^2	RMSE	d			
3 AQEs, 3 Temp, 3 Humidity	CO	2	Sigmoid	0.005	1000	0.89	0.96	0.21	0.72	0.9	(0,1,5). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	2	Sigmoid	0.005	1000	0.88	0.96	0.2	0.72	0.9	(1,1,1). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	2	Sigmoid	0.005	10000	0.88	0.96	0.2	0.72	0.9	(1,1,1). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	3	Sigmoid	0.005	100	0.9	0.97	0.21	0.73	0.9	(1,1,1). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	5	Tanh	0.05	50000	0.694	0.97	0.64	0.31	0.9	(1,1,1). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	5	Sigmoid	0.0005	5000	0	0.65	106.67	0.11	0.9	(1,1,1). DDCN. CSS.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	6	Sigmoid	0.05	10	0.48	0.66	18.14	0.45	0.9	(1,1,1). DDCN. ML.	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	7	Tanh	0.00001	100	0	0.65	20.21	0.42	0.9	(1,1,1). DDCN. CSS.	WEEKLY

Table 5.8 Selected result of single hidden layer with non-seasonal ARIMA estimator

INDEPENDENT VARIABLES	TARGET	NUMBER OF NODES (HIDDEN LAYER)	OUTPUT FUNCTION	LEARNING PARAMETER	EPOCH	TRAINING PHASE R^2	TESTING PHASE			TRAINING PERCENTAGE	ARIMA ESTIMATOR & DECISION SCHEME	BATCHES
							R^2	RMSE	d			
3 AQEs, 3 Temp, 3 Humidity	CO	3,6,9 (3)	Tanh	0.001	50000	0.86	0.95	0.08	0.93	0.9	(0,1,5). DDCN. ML	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	3,6,9 (3)	Logistic	0.05	100	0	0.95	0.22	0.46	0.9	(0,1,5). DDCN. ML	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	3,6,9 (3)	Logistic	0.05	50000	0.93	0.95	0.07	0.96	0.9	(0,1,5). DDCN. ML	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	CO	3,6,9 (3)	Tanh	0.001	50000	0.82	0.95	0.05	0.97	0.9	(1,1,1). DDCN. ML	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	3,6,9(3)	Logistic	0.005	1	0.01	0.93	25.28	0.36	0.9	(1,1,1). DDCAN. CSS	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	3,6,9(3)	Logistic	0.0001	10	0.21	0.92	25.48	0.36	0.9	(1,1,1). DDCAN. CSS	WEEKLY
3 AQEs, 3 Temp, 3 Humidity	NO_2	9,6,3(3)	logistic	0.01	10	0.34	0.93	25.24	0.36	0.9	(1,1,1). DDCAN. CSS	WEEKLY

Table 5.9 Selected result of multi-layer perceptron with non-seasonal ARIMA estimator

5.5 Summary

The adjustment module is discussed in this chapter. Three methods were tested within the module: LR, MLR, and ANN. LR and MLR use statistics in fitting the reference, while ANN is the further development technique from statistics [96]. To rely solely on the data from one AQE to fit the ECan is not considered to be good practice because the LR had performed very poorly. The MLR does not significantly improve the fitting performance, even though other independent variables from adjacent sensors are considered. ANN outperforms LR and MLR.

There are many network types in the ANN. We tested three that seem to be the most commonly used ones in the gas sensor context: single hidden layer (SHL), gradient boosting (GB), and multi-layer perceptron (MLP). Each method was trained, evaluated, and tested using two data sources from the AQE and ECan. *R-square*, RMSE, and *d*-value were used to evaluate the fitting performance of all methods in the testing phase.

The effect of the outlier module to the adjustment module had also been conducted and indicated an interesting finding. Although the use of the outlier module with seasonal ARIMA estimators has a slight increase in the calibration of NO_2 readings, the adjustment module has only utilised 26% of rows (540 out of about 2000 rows) from the outlier module. The use of the outlier module with non-seasonal ARIMA estimators has a good impact to the adjustment module since it can increase the AQE readings to fitting both CO and NO_2 reference gas of up to 90% using *R-square* in the testing phase. However, the dataset has also reduced significantly by the outlier module when passing it to the adjustment module. More work is needed to investigate further about this interesting finding, particularly to address the missing data presenting to the users. Without the presence of the outlier module, the

AQE CO sensors can fit the ECan by 80%, while the NO_2 sensors can fit 40% at the right network configuration.

The notion of our work is in the use of redundant adjacent sensor nodes where it can fit the output of the three AQEs with the ECan using ANN. Redundant sensor nodes help the ANN to increase the accuracy of the sensors reading because it adds more features to the network. ANN is known working at best with a lot of inputs, so redundant sensors help the ANN to make better prediction and increase the fitting process.

Chapter 6: Conclusion

This thesis has presented the design and implementation to increase the accuracy of three Air Quality Eggs (AQEs) in a supervised manner. Two modules were proposed and evaluated during the training and testing stages. First, the interesting findings are outlined, including answers to the research questions. Finally, further possible work is discussed.

6.1 Summary of Findings

An assessment of the statistical properties of the low-cost sensor readings enabled calibration of the sensors. The temperature, humidity and CO sensors had an index of agreement (*d*) of 0.98, 0.98, and 0.8, respectively, among the three AQEs. This indicated that the sensors had reliable readings. The NO_2 sensors, on the other hand, were not reliable, as the index of agreement was 0.4. The analysis of variance (ANOVA) on the 5-second readings test statistically proved that the four sensors on the three AQEs were different. The ANOVA test on the averaged one-hour readings indicated that AQE1 had the most deviated readings of the AQEs. Then, Tukey's test on the 5-second readings showed the quantitative difference between sensors on the AQEs. The biggest variation between sensors was: (i) NO_2 sensors on AQE1 and AQE2 by 16.8 ppb, (ii) temperature sensors on AQE1 and AQE2 by 0.46 degrees Celsius, (iii) humidity sensors on AQE1 and AQE2 by 2.25%, and (iv) CO sensors on AQE2 and AQE3 by 0.08 ppm.

This study utilised on-site redundant sensors and supervised learning to calibrate the low-cost AQE sensors. Two modules were proposed for calibration: outlier and adjustment modules. The outlier module aims to filter the sensors' readings from outliers, while the

adjustment module is there to increase the readings accuracy. Early evaluation of the effect of the outlier module on the adjustment module showed that the outlier module can enhance the fitting of the adjustment module to the reference using the non-seasonal ARIMA estimators, and either DDCN or DDCAN as the decision scheme, but, with the compromise of a decrease in the number of rows in the dataset. Therefore, at this point it is clear that more data and further study are needed to fully use the outlier module.

ARIMA estimators were employed in the four proposed detection schemes on the outlier module. The four proposed schemes were: Static Detection (SD), Dynamic Detection (DD), Dynamic Detection with Comparison to Neighbour (DDCN), and Dynamic Detection with Comparison to ARIMA's Neighbour (DDCAN). DDCN and DDCAN had a good accuracy of nearly 100%, particularly for the first-time-use sensors. These two schemes outperformed the other two schemes. It is interesting that the selection of parameter estimation for an ARIMA estimator should be chosen carefully since the calculation may fail to converge. Examining the sensors' difference can help in choosing the ARIMA estimator. The findings of this study strengthen Dent's suggestion on the use of the Maximum Likelihood (ML) or Conditional Sum-of-Square (CSS) method in estimating parameters. CSS was better suited to the monthly data batch, with more of the dataset being used, while ML was more appropriate for the weekly batch (fewer rows).

Again, using the statistical properties between the sensors helps in determining which outlier method to be employed. DDCN can perform at nearly a 100% accuracy in detecting outliers for small different readings among adjacent sensors. Also, DDCN is likely to consume less computational power than DDCAN. For a moderate to large different readings case, DDCAN seems more suitable. Thus, it appears that DDCN works best to detect an outlier in

temperature, humidity, and CO sensors. Meanwhile, we can employ DDCAN for detecting outliers in NO_2 sensors.

The three research questions central to this study were asked in chapter 3. These questions can now be addressed and conclusions drawn.

Q1: How much do readings vary between different AQE sensor products?

The ANOVA test indicates the difference readings between sensors.

The Tukey test further pinpoints the quantitative readings difference between the sensors on the three AQEs. We maintain that the readings difference between the corresponding sensors is still acceptable where the CO, temperature and humidity sensors differ by less than 0.1 ppm, 0.5 degrees Celsius, and 2.5%, respectively. The NO_2 readings on the AQEs, on the other hand, varied at 16.84 ppb which can be considered as a relatively moderate difference which cannot be tolerated.

Q2. By adding redundant sensors, can the outlier module identify outliers?

This study does not have a sufficient dataset to conclude on this question. The ECan data is used as a reference, but the data cannot be used to determine if the dataset has outliers or not. A pre-defined dataset classifying outliers and normal data would be needed to support the ECan data. However, if we assume that the AQEs are brand new, with no outliers on the dataset, we can infer that the outlier module can identify outliers, excepting the NO_2 sensors. If this assumption is made, then it can be said that in the testing phase DDCN and DDCAN can successfully identify outlier with an accuracy of up to 100% on specific scenarios. It is important to note that although filtering the NO_2 sensors in the outlier module has a good performance, the readings may contain unidentified outliers, since the index of

agreement between the NO_2 sensors is less than 50%. In this case, our assumption may not apply for the NO_2 readings.

Q3. Could we find a mathematical model for AQE in the adjustment module that can fit a reference device?

Yes, we could. Three methods were tested and evaluated on the adjustment module: Linear Regression (LR), Multi Linear Regression (MLR), and Artificial Neural Network (ANN). LR and MLR cannot fit the ECan data in the testing phase since its *R-square* is less than 0.05 (5%). ANNs perform better in the testing phase; three ANN topologies were examined: Single Hidden Layer (SHL), Gradient Boosting (GB), and Multi-Layer Perceptrons (MLP). The GB network has the best performance when the outlier module does not present. Relying on the reading of the sensors and their adjacent nodes, the CO adjustment using GB achieves *R-square* of 0.8 (80%) with small RMSE of 0.2, while NO_2 adjustment reaches *R-square* of 0.4 (40%) with the risk of a high variance on the result as indicated by a high RMSE rate. The high RMSE rate indicates the possibility that the ANN output prediction may return varied results for the same network configuration. Passing normalized inputs to the ANN can result in low RMSE rate, but a one-year sensor dataset is required.

The deployment of the adjustment module using the ANN and the outlier module using non-seasonal ARIMA estimators shows an interesting finding. The presence of the outlier module before the adjustment module may boost the calibration process, but more experimentation and data are needed since the dataset used in the experiment is limited. Another interesting finding is that the use of redundant sensors helps the ANN to learn better about the dataset and make a better prediction.

6.2 Future Work

It is interesting that the prediction of the CO and NO_2 reference using the corresponding AQE sensors, with or without the outlier module, is around 80%. This early finding requires further experiment, particularly because of the limited dataset. We suggest the following dataset to be included in future work:

(i) A full one-year dataset from the AQEs.

The availability of this data would potentially alleviate problems due to the stochastic process. The use of a one-year dataset might add the use of data normalization to reduce high RMSE in the NO_2 sensors. In addition, having one year of data might add variables such as a seasonality factor to the ANN, because the algorithm could work with a larger dataset and features.

(ii) The number of redundant AQEs

Having more than three AQEs on a site or within a certain range could increase the understanding of the dataset in the outlier module and the ANN performance in the adjustment module.

(iii). Other types of dataset

This study assumes there is no outlier in the dataset. Features such as wind speed, number of vehicles crossing on the site, and ozone concentration could be added to the explanatory variables so that the outliers could easily be distinguished from the dataset.

Reference

[1] W. H. Organization, “Health aspects of air pollution: results from the WHO project" Systematic review of health aspects of air pollution in Europe",” 2004.

[2] N. Castell, F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova, “Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?,” *Environment International*, 2016.

[3] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?." pp. 11-16.

[4] W. Device. "Air Quality Egg," 22 April 2016, 2016; http://www.airqualityegg.com/.

[5] D. Demuth, D. Nuest, A. Bröring, and E. Pebesma, "The AirQuality SenseBox." p. 5146.

[6] SenseMaker. "Air Quality Egg by SenseMaker," 23 April 2016, 2016; https://www.kickstarter.com/projects/edborden/air-quality-egg.

[7] S. Choi, N. Kim, H. Cha, and R. Ha, “Micro Sensor Node for Air Pollutant Monitoring: Hardware and Software Issues,” *Sensors,* vol. 9, no. 10, 2009.

[8] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost available sensors for air quality monitoring. Part A: Ozone and nitrogen dioxide,” *Sensors and Actuators B: Chemical,* vol. 215, pp. 249-257, 8//, 2015.

[9] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. Part B: NO, CO and CO 2,” *Sensors and Actuators B: Chemical,* vol. 238, pp. 706-715, 2017.

[10] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, “On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario,” *Sensors and Actuators B: Chemical,* vol. 129, no. 2, pp. 750-757, 2008.

[11] S. De Vito, M. Piga, L. Martinotto, and G. Di Francia, “CO, NO 2 and NO x urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization,” *Sensors and Actuators B: Chemical,* vol. 143, no. 1, pp. 182-191, 2009.

[12] L. Inc. "Xively by LogMeIn," 6 May 2016, 2016; http://www.xively.com/.

[13] OpenSensors.io. "OpenSensors.io - Connecting Things," 6 May 2016, 2016; http://www.opensensors.io/.

[14] MQTT. "FAQ - Frequently Asked Questions," 7 April 2017, 2017; http://mqtt.org/faq.

[15] R. D. Peng, F. Dominici, R. Pastor-Barriuso, S. L. Zeger, and J. M. Samet, "Seasonal analyses of air pollution and mortality in 100 US cities," *American journal of epidemiology,* vol. 161, no. 6, pp. 585-594, 2005.

[16] V. J. Hodge, and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review,* vol. 22, no. 2, pp. 85-126, 2004.

[17] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[18] W. H. Organization, "Health aspects of air pollution with particulate matter, ozone and nitrogen dioxide: report on a WHO working group, Bonn, Germany 13-15 January 2003," 2003.

[19] N. Kularatna, and B. H. Sudantha, "An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements," *IEEE Sensors Journal,* vol. 8, no. 4, pp. 415-422, 2008.

[20] J. M. Johnson, and M. Center, *The Cost of Regulations Implementing the Clean Water Act*: Mercatus Center, George Mason University, 2004.

[21] G. M. Lovett, D. A. Burns, C. T. Driscoll, J. C. Jenkins, M. J. Mitchell, L. Rustad, J. B. Shanley, G. E. Likens, and R. Haeuber, "Who needs environmental monitoring?," *Frontiers in Ecology and the Environment,* vol. 5, no. 5, pp. 253-260, 2007.

[22] New Zealand Institute of Chemistry, "Air Pollution Monitoring," *Environment*, New Zealand Institute of Chemistry, 2008.

[23] E. J. Nicholas, "The Betweenness of Place: Towards a Geography of Modernity," *Baltimore Johns Hopkins University*, 1991.

[24] D. C. Korten, "The management of social transformation," *Public Administration Review,* vol. 41, no. 6, pp. 609-618, 1981.

[25] L. E. Kruger, and M. A. Shannon, "Getting to know ourselves and our places through participation in civic social assessment," *Society & Natural Resources,* vol. 13, no. 5, pp. 461-478, 2000.

[26] M. Kerr, E. Ely, V. Lee, and A. Mayio, “A profile of volunteer environmental monitoring: National survey results,” *Lake and Reservoir Management,* vol. 9, no. 1, pp. 1-4, 1994.

[27] C. T. Conrad, and T. Daoust, “Community-based monitoring frameworks: Increasing the effectiveness of environmental stewardship,” *Environmental Management,* vol. 41, no. 3, pp. 358-366, 2008.

[28] C. C. Conrad, and K. G. Hilchey, “A review of citizen science and community-based environmental monitoring: issues and opportunities,” *Environmental monitoring and assessment,* vol. 176, no. 1-4, pp. 273-291, 2011.

[29] B. Savan, A. J. Morgan, and C. Gore, “Volunteer environmental monitoring and the role of the universities: the case of Citizens' Environment Watch,” *Environmental management,* vol. 31, no. 5, pp. 0561-0568, 2003.

[30] D. W. Dockery, C. A. Pope, X. Xu, J. D. Spengler, J. H. Ware, M. E. Fay, B. G. Ferris Jr, and F. E. Speizer, “An association between air pollution and mortality in six US cities,” *New England journal of medicine,* vol. 329, no. 24, pp. 1753-1759, 1993.

[31] K. Katsouyanni, G. Touloumi, E. Samoli, A. Gryparis, A. Le Tertre, Y. Monopolis, G. Rossi, D. Zmirou, F. Ballester, and A. Boumghar, “Confounding and effect modification in the short-term effects of ambient particles on total mortality: results from 29 European cities within the APHEA2 project,” *Epidemiology,* vol. 12, no. 5, pp. 521-531, 2001.

[32] A. Abelsohn, M. D. Sanborn, B. J. Jessiman, and E. Weir, “Identifying and managing adverse environmental health effects: 6. Carbon monoxide poisoning,” *Canadian Medical Association Journal,* vol. 166, no. 13, pp. 1685-1690, 2002.

[33] R. Bogue, “Recent developments in MEMS sensors: A review of applications, markets and technologies,” *Sensor Review,* vol. 33, no. 4, pp. 300-304, 2013.

[34] A. Vasiliev, A. Sokolov, A. Legin, N. Samotaev, K. Y. Oblov, V. Kim, S. Tkachev, S. Gubin, G. Potapov, and Y. V. Kokhtina, “Additive Technologies for Ceramic MEMS Sensors,” *Procedia Engineering,* vol. 120, pp. 1087-1090, 2015.

[35] *Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on Ambient Air Quality and Cleaner Air for Europe* E. Union Standard OJ L 152*,* 2008.

[36] S. De Vito, A. Castaldo, F. Loffredo, E. Massera, T. Polichetti, I. Nasti, P. Vacca, L. Quercia, and G. Di Francia, “Gas concentration estimation in ternary mixtures with

room temperature operating sensor array using tapped delay architectures," *Sensors and Actuators B: Chemical,* vol. 124, no. 2, pp. 309-316, 2007.

[37] M. Kamionka, P. Breuil, and C. Pijolat, "Calibration of a multivariate gas sensing device for atmospheric pollution measurement," *Sensors and Actuators B: Chemical,* vol. 118, no. 1, pp. 323-327, 2006.

[38] E. F. K. Aguiar, H. L. Roig, L. H. Mancini, and E. N. C. B. de Carvalho, "Low-Cost Sensors Calibration for Monitoring Air Quality in the Federal District—Brazil," *Journal of Environmental Protection,* vol. 6, no. 02, pp. 173, 2015.

[39] N. Masson, R. Piedrahita, and M. Hannigan, "Approach for quantification of metal oxide type semiconductor gas sensors used for ambient air quality monitoring," *Sensors and Actuators B: Chemical,* vol. 208, pp. 339-345, 2015.

[40] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B: Chemical,* vol. 110, no. 2, pp. 304-311, 2005.

[41] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele, "Participatory air pollution monitoring using smartphones," *Mobile Sensing*, pp. 1-5, 2012.

[42] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors." pp. 349-350.

[43] C. S. Team. "Common Sense - Mobile Sensing for Community Action," 29 June 2016, 2016; http://www.communitysensing.org/index.php.

[44] R. Honicky, E. A. Brewer, E. Paulos, and R. White, "N-smarts: networked suite of mobile atmospheric real-time sensors." pp. 25-30.

[45] Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang, "Maqs: A mobile sensing system for indoor air quality." pp. 493-494.

[46] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting,* vol. 14, no. 1, pp. 35-62, 1998.

[47] S. Lek, and J.-F. Guégan, "Artificial neural networks as a tool in ecological modelling, an introduction," *Ecological modelling,* vol. 120, no. 2, pp. 65-73, 1999.

[48] M. W. Gardner, and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment,* vol. 32, no. 14, pp. 2627-2636, 1998.

[49] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *IEEE computer,* vol. 29, no. 3, pp. 31-44, 1996.

[50] M. Gardner, and S. Dorling, “Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London,” *Atmospheric Environment,* vol. 33, no. 5, pp. 709-719, 1999.

[51] S. S. Nagendra, and M. Khare, “Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions,” *Ecological Modelling,* vol. 190, no. 1, pp. 99-115, 2006.

[52] G. Huyberechts, P. Szecowka, J. Roggen, and B. Licznerski, “Simultaneous quantification of carbon monoxide and methane in humid air using a sensor array and an artificial neural network,” *Sensors and Actuators B: Chemical,* vol. 45, no. 2, pp. 123-130, 1997.

[53] M. Kamionka, P. Breuil, and C. Pijolat, “Atmospheric pollution measurement with a multi-materials sensing device,” *Materials Science and Engineering: C,* vol. 26, no. 2, pp. 290-296, 2006.

[54] A. Szczurek, P. Szecowka, and B. Licznerski, “Application of sensor array and neural networks for quantification of organic solvent vapours in air,” *Sensors and Actuators B: Chemical,* vol. 58, no. 1, pp. 427-432, 1999.

[55] M. Pardo, and G. Sberveglieri, “Comparing the performance of different features in sensor arrays,” *Sensors and Actuators B: Chemical,* vol. 123, no. 1, pp. 437-443, 2007.

[56] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *IEEE Communications Surveys & Tutorials,* vol. 12, no. 2, pp. 159-170, 2010.

[57] V. Bamnett, and T. Lewis, “Outliers in statistical data,” 1994.

[58] D. M. Hawkins, *Identification of outliers*: Springer, 1980.

[59] J. Neyman, and E. L. Scott, “Outlier proneness of phenomena and of related distributions,” *Optimizing methods in statistics*, pp. 413-423, 1971.

[60] R. F. Green, “Outlier-prone and outlier-resistant distributions,” *Journal of the American Statistical Association,* vol. 71, no. 354, pp. 502-505, 1976.

[61] M. Markou, and S. Singh, “Novelty detection: a review—part 1: statistical approaches,” *Signal processing,* vol. 83, no. 12, pp. 2481-2497, 2003.

[62] M. Markou, and S. Singh, “Novelty detection: a review—part 2:: neural network based approaches,” *Signal processing,* vol. 83, no. 12, pp. 2499-2521, 2003.

[63] I. Chang, G. C. Tiao, and C. Chen, “Estimation of time series parameters in the presence of outliers,” *Technometrics,* vol. 30, no. 2, pp. 193-204, 1988.

[64] V. R. Prybutok, J. Yi, and D. Mitchell, “Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations,” *European Journal of Operational Research,* vol. 122, no. 1, pp. 31-40, 2000.

[65] OpenSensors.io, "Getting Started," https://opensensorsio.helpscoutdocs.com/category/6-getting-started, [21 July 2016, 2016].

[66] W. Devices. "Hardware-Sensors," 20 July 2016, 2016; airqualityegg.wikispaces.com/Hardware-Sensors.

[67] e2v, "MiCS Application Note 2 FAQ," http://airqualityegg.wikispaces.com/file/view/mics_an2 - MICS FAQ.pdf/335989698/mics_an2 - MICS FAQ.pdf, [20 July 2016, 2012].

[68] A. Electronics, "Digital-output relative humidity & temperature sensor/module AM2303," http://airqualityegg.wikispaces.com/file/view/DHT22.pdf/357156114/DHT22.pdf, [20 July 2016.

[69] Adafruit, "Humidity: Adafruit Industries, Unique & fun DIY electronics and kits," https://www.adafruit.com/category/35_66, [20 July 2016, 2016].

[70] E. Canterbury. "Data Sets. ECan Data Catalogue," 6 December 2016, 2016; http://data.ecan.govt.nz/Catalogue/Search?CollectionId=4.

[71] W. W.-S. Wei, *Time series analysis*: Addison-Wesley publ Reading, 1994.

[72] R. H. Shumway, and D. S. Stoffer, *Time series analysis and its applications: with R examples*: Springer Science & Business Media, 2010.

[73] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*: John Wiley & Sons, 2015.

[74] W. N. R. Venables, B. D., *Modern Applied Statistics with S. Fourth Edition*: Springer, New York, 2002.

[75] W. Dent, and A.-S. Min, "A Monte Carlo study of autoregressive integrated moving average processes," *Journal of Econometrics,* vol. 7, no. 1, pp. 23-55, 1978.

[76] C. F. Ansley, and P. Newbold, "Finite sample properties of estimators for autoregressive moving average models," *Journal of Econometrics,* vol. 13, no. 2, pp. 159-183, 1980.

[77] S. C. Hillmer, and G. C. Tiao, "Likelihood function of stationary multiple autoregressive moving average models," *Journal of the American Statistical Association,* vol. 74, no. 367, pp. 652-660, 1979.

[78] D. R. Osborn, "On the criteria functions used for the estimation of moving average processes," *Journal of the American Statistical Association,* vol. 77, no. 378, pp. 388-392, 1982.

[79] G. Gardner, A. C. Harvey, and G. D. Phillips, "Algorithm AS 154: An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 29, no. 3, pp. 311-322, 1980.

[80] J. Durbin, and S. J. Koopman, *Time series analysis by state space methods*: OUP Oxford, 2012.

[81] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, "Akaike information criterion statistics," *Dordrecht, The Netherlands: D. Reidel*, 1986.

[82] N. Z. T. Guide. "New Zealand Weather and Climate," 16 August 2016, 2016; http://www.tourism.net.nz/new-zealand/about-new-zealand/weather-and-climate.html.

[83] C. J. Willmott, "On the validation of models," *Physical geography,* vol. 2, no. 2, pp. 184-194, 1981.

[84] J. Chambers, A. Freeny, and R. Heiberger, "Analysis of variance; designed experiments," *Statistical models in S*, pp. 145-193, 1992.

[85] B. S. Yandell, *Practical data analysis for designed experiments*: Crc Press, 1997.

[86] A. M. Brown, "A new software for carrying out one-way ANOVA post hoc tests," *Computer methods and programs in biomedicine,* vol. 79, no. 1, pp. 89-95, 2005.

[87] S. Overflow. "TukeyHSD adjusted P value is 0.0000000," 7 February 2017, 2017; http://stackoverflow.com/questions/16470404/tukeyhsd-adjusted-p-value-is-0-0000000.

[88] S. Exchange. "Errors in optim when fitting arima model in R," 8 February 2017, 2017; http://stats.stackexchange.com/questions/84330/errors-in-optim-when-fitting-arima-model-in-r.

[89] C. R. Rao, and H. Toutenburg, "Linear models," *Linear models*, pp. 3-18: Springer, 1995.

[90] G. R. w. c. f. others. "gbm: Generalized Boosted Regression Models," http://cran.r-project.org/package=gbm.

[91] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, pp. 172-181, 1999.

[92] G. Ridgeway, "Generalized Boosted Models: A guide to the gbm package," *Update,* vol. 1, no. 1, pp. 2007, 2007.

[93] C. B. a. J. M. Benitez, "Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS," *Journal of Statistical Software,* vol. 46, no. 7, pp. 1-26, 2012.

[94] A. C. Comrie, "Comparing neural networks and regression models for ozone forecasting," *Journal of the Air & Waste Management Association,* vol. 47, no. 6, pp. 653-663, 1997.

[95] D. R. Legates, and G. J. McCabe, "Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation," *Water resources research,* vol. 35, no. 1, pp. 233-241, 1999.

[96] I. Basheer, and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods,* vol. 43, no. 1, pp. 3-31, 2000.

[97] N. Murata, S. Yoshizawa, and S.-i. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks,* vol. 5, no. 6, pp. 865-872, 1994.

Appendix A Testing Scenario for Outlier Module

The following page contains the result of running various scenarios for outlier module using monthly and weekly data sequence. Note that the red colour font indicates the calculation using CSS for estimating parameters. A dash means the calculation cannot converge.

Monthly NO2

Method	AQEs	2			3		
		TP	FN	Accuracy	TP	FN	Accuracy
(1, 1, 1) x (12, 1, 12)							
SD	AQE1	631	28	0.958	535	29	0.949
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(1, 1, 1) x (24, 1, 24)							
SD	AQE1	629	30	0.954	534	30	0.947
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
SD	AQE1	630	29	0.956	532	2	0.996
Forecast ± σ	AQE2	301	357	0.457	301	263	0.534
	AQE3	406	252	0.617	277	287	0.491
(0,1,5)							
SD	AQE1	628	31	0.953	532	32	0.943
Forecast ± σ	AQE2	471	187	0.716	388	176	0.688
	AQE3	362	296	0.550	333	231	0.590
(1,1,1)							
SD	AQE1	628	31	0.953	532	32	0.943
Forecast ± σ	AQE2	484	174	0.736	396	168	0.702
	AQE3	356	302	0.541	326	238	0.578
(1, 1, 1) x (12, 1, 12)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-

(1, 1, 1) x (24, 1, 24)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	561	97	0.853	531	33	0.941
	AQE3	604	54	0.918	516	48	0.915
(0,1,5)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	653	5	0.992	554	10	0.982
	AQE3	554	104	0.842	487	77	0.863
(1,1,1)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	552	106	0.839	486	78	0.862
(1, 1, 1) x (12, 1, 12)							
SD	AQE1	428	231	0.649	448	116	0.794
Forecast ± σ	AQE2	369	289	0.561	346	218	0.613
	AQE3	303	355	0.460	285	279	0.505
(1, 1, 1) x (24, 1, 24)							
SD	AQE1	383	276	0.581	332	232	0.589
Forecast ± σ	AQE2	188	470	0.286	184	380	0.326
	AQE3	344	314	0.523	311	253	0.551
(0, 1, 5) x (12, 1, 12)							
SD	AQE1	448	211	0.680	461	103	0.817
Forecast ± σ	AQE2	388	270	0.590	360	204	0.638
	AQE3	352	306	0.535	321	243	0.569
(0,1,5)							
SD	AQE1	628	31	0.953	532	32	0.943
Forecast ± σ	AQE2	471	187	0.716	387	177	0.686
	AQE3	362	296	0.550	333	231	0.590
(1,1,1)							
SD	AQE1	628	31	0.953	532	32	0.943
Forecast ± σ	AQE2	484	174	0.736	396	168	0.702
	AQE3	356	302	0.541	326	238	0.578
(1, 1, 1) x (12, 1, 12)							
SD	AQE1	631	28	0.958	550	14	0.975
Forecast ± 2σ	AQE2	621	37	0.944	546	18	0.968
	AQE3	509	149	0.774	465	99	0.824

(1, 1, 1) x (24, 1, 24)							
SD	AQE1	511	148	0.775	480	84	0.851
Forecast ± 2σ	AQE2	369	289	0.561	425	139	0.754
	AQE3	539	119	0.819	475	89	0.842
(0, 1, 5) x (12, 1, 12)							
SD	AQE1	633	26	0.961	550	14	0.975
Forecast ± 2σ	AQE2	631	27	0.959	550	14	0.975
	AQE3	546	112	0.830	480	84	0.851
(0,1,5)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	653	5	0.992	554	10	0.982
	AQE3	554	104	0.842	487	77	0.863
(1,1,1)							
SD	AQE1	653	6	0.991	557	7	0.988
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	552	106	0.839	486	78	0.862
(1, 1, 1) x (12, 1, 12)							
DD	AQE1	491	24	0.953	528	36	0.936
Forecast ± σ	AQE2	299	359	0.454	408	156	0.723
	AQE3	-	-	-	-	-	-
(1, 1, 1) x (24, 1, 24)							
DD	AQE1	490	25	0.951	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
DD	AQE1	491	24	0.953	537	27	0.952
Forecast ± σ	AQE2	301	357	0.457	-	-	-
	AQE3	403	255	0.612	439	125	0.778
(0,1,5)							
DD	AQE1	488	27	0.948	528	36	0.936
Forecast ± σ	AQE2	471	187	0.716	405	159	0.718
	AQE3	362	296	0.550	414	150	0.734
(1,1,1)							
DD	AQE1	628	31	0.953	516	48	0.915
Forecast ± σ	AQE2	484	174	0.736	421	143	0.746
	AQE3	356	302	0.541	444	120	0.787
(1, 1, 1) x (12, 1, 12)							
DD	AQE1	509	6	0.988	556	8	0.986
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-

(1, 1, 1) x (24, 1, 24)							
DD	AQE1	510	5	0.990	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
DD	AQE1	509	6	0.988	557	7	0.988
Forecast ± 2σ	AQE2	561	97	0.853	-	-	-
	AQE3	604	54	0.918	550	14	0.975
(0,1,5)							
DD	AQE1	509	6	0.988	556	8	0.986
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	554	104	0.842	558	6	0.989
(1,1,1)							
DD	AQE1	653	6	0.991	554	10	0.982
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	552	106	0.839	553	11	0.980
(1, 1, 1) x (12, 1, 12)							
DD	AQE1	428	231	0.649	496	68	0.879
Forecast ± σ	AQE2	369	289	0.561	347	217	0.615
	AQE3	303	355	0.460	389	175	0.690
(1, 1, 1) x (24, 1, 24)							
DD	AQE1	383	276	0.581	477	87	0.846
Forecast ± σ	AQE2	188	470	0.286	101	463	0.179
	AQE3	344	314	0.523	387	177	0.686
(0, 1, 5) x (12, 1, 12)							
DD	AQE1	448	211	0.680	490	74	0.869
Forecast ± σ	AQE2	388	270	0.590	326	238	0.578
	AQE3	352	306	0.535	303	261	0.537
(0,1,5)							
DD	AQE1	628	31	0.953	517	47	0.917
Forecast ± σ	AQE2	471	187	0.716	406	158	0.720
	AQE3	362	296	0.550	414	150	0.734
(1,1,1)							
DD	AQE1	628	31	0.953	516	48	0.915
Forecast ± σ	AQE2	484	174	0.736	418	146	0.741
	AQE3	356	302	0.541	411	153	0.729
(1, 1, 1) x (12, 1, 12)							
DD	AQE1	631	28	0.958	552	12	0.979
Forecast ± 2σ	AQE2	621	37	0.944	540	24	0.957
	AQE3	509	149	0.774	548	16	0.972

(1, 1, 1) x (24, 1, 24)							
DD	AQE1	511	148	0.775	548	16	0.972
Forecast ± 2σ	AQE2	369	289	0.561	322	242	0.571
	AQE3	539	119	0.819	558	6	0.989
(0, 1, 5) x (12, 1, 12)							
DD	AQE1	633	26	0.961	548	16	0.972
Forecast ± 2σ	AQE2	631	27	0.959	532	32	0.943
	AQE3	546	112	0.830	517	47	0.917
(0,1,5)							
DD	AQE1	653	6	0.991	554	10	0.982
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	554	104	0.842	558	6	0.989
(1,1,1)							
DD	AQE1	653	6	0.991	554	10	0.982
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	552	106	0.839	556	8	0.986
(1, 1, 1) x (12, 1, 12)							
DDCN	AQE1	522	137	0.792	511	53	0.906
Forecast ± σ	AQE2	424	234	0.644	354	210	0.628
	AQE3	463	195	0.704	442	122	0.784
(1, 1, 1) x (24, 1, 24)							
DDCN	AQE1	467	192	0.709	498	66	0.883
Forecast ± σ	AQE2	284	374	0.432	158	406	0.280
	AQE3	488	170	0.742	487	77	0.863
(0, 1, 5) x (12, 1, 12)							
DDCN	AQE1	539	120	0.818	507	57	0.899
Forecast ± σ	AQE2	444	214	0.675	331	233	0.587
	AQE3	495	163	0.752	392	172	0.695
(0,1,5)							
DDCN	AQE1	648	11	0.983	524	40	0.929
Forecast ± σ	AQE2	537	121	0.816	465	99	0.824
	AQE3	501	157	0.761	494	70	0.876
(1,1,1)							
DDCN	AQE1	648	11	0.983	523	41	0.927
Forecast ± σ	AQE2	553	105	0.840	491	73	0.871
	AQE3	496	162	0.754	475	89	0.842
(1, 1, 1) x (12, 1, 12)							
DDCN	AQE1	657	2	0.997	556	8	0.986
Forecast ± 2σ	AQE2	628	30	0.954	544	20	0.965
	AQE3	651	7	0.989	562	2	0.996

(1, 1, 1) x (24, 1, 24)							
DDCN	AQE1	623	36	0.945	555	9	0.984
Forecast ± 2σ	AQE2	494	164	0.751	456	108	0.809
	AQE3	654	4	0.994	564	0	1.000
(0, 1, 5) x (12, 1, 12)							
DDCN	AQE1	657	2	0.997	555	9	0.984
Forecast ± 2σ	AQE2	634	24	0.964	540	24	0.957
	AQE3	656	2	0.997	561	3	0.995
(0,1,5)							
DDCN	AQE1	658	1	0.998	558	6	0.989
Forecast ± 2σ	AQE2	654	4	0.994	556	8	0.986
	AQE3	656	2	0.997	564	0	1.000
(1,1,1)							
DDCN	AQE1	658	1	0.998	558	6	0.989
Forecast ± 2σ	AQE2	654	4	0.994	556	8	0.986
	AQE3	656	2	0.997	563	1	0.998
(1, 1, 1) x (12, 1, 12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(1, 1, 1) x (24, 1, 24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
DDCAN	AQE1	659	0	1.000	-	-	-
Forecast ± σ	AQE2	333	325	0.506	-	-	-
	AQE3	600	58	0.912	-	-	-
(0,1,5)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	540	118	0.821	465	99	0.824
	AQE3	655	3	0.995	560	4	0.993
(1,1,1)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	562	96	0.854	494	70	0.876
	AQE3	655	3	0.995	563	1	0.998
(1, 1, 1) x (12, 1, 12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-

(1, 1, 1) x (24, 1, 24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0, 1, 5) x (12, 1, 12)							
DDCAN	AQE1	659	0	1.000	-	-	-
Forecast ± σ	AQE2	561	97	0.853	-	-	-
	AQE3	655	3	0.995	-	-	-
(0,1,5)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	658	0	1.000	564	0	1.000
(1,1,1)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	658	0	1.000	564	0	1.000
(1, 1, 1) x (12, 1, 12)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	412	246	0.626	353	211	0.626
	AQE3	640	18	0.973	551	13	0.977
(1, 1, 1) x (24, 1, 24)							
DDCAN	AQE1	658	1	0.998	546	18	0.968
Forecast ± σ	AQE2	212	446	0.322	210	354	0.372
	AQE3	418	240	0.635	488	76	0.865
(0, 1, 5) x (12, 1, 12)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	436	222	0.663	328	236	0.582
	AQE3	647	11	0.983	543	21	0.963
(0,1,5)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	540	118	0.821	466	98	0.826
	AQE3	655	3	0.995	560	4	0.993
(1,1,1)							
DDCAN	AQE1	659	0	1.000	564	0	1.000
Forecast ± σ	AQE2	562	96	0.854	491	73	0.871
	AQE3	655	3	0.995	563	1	0.998
(1, 1, 1) x (12, 1, 12)							
DDCAN	AQE1	631	28	0.958	552	12	0.979
Forecast ± 2σ	AQE2	621	37	0.944	540	24	0.957
	AQE3	509	149	0.774	548	16	0.972

(1, 1, 1) x (24, 1, 24)							
DDCAN	AQE1	511	148	0.775	548	16	0.972
Forecast ± 2σ	AQE2	369	289	0.561	322	242	0.571
	AQE3	539	119	0.819	558	6	0.989
(0, 1, 5) x (12, 1, 12)							
DDCAN	AQE1	633	26	0.961	548	16	0.972
Forecast ± 2σ	AQE2	631	27	0.959	532	32	0.943
	AQE3	546	112	0.830	517	47	0.917
(0,1,5)							
DDCAN	AQE1	653	6	0.991	554	10	0.982
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	554	104	0.842	558	6	0.989
(1,1,1)							
DDCAN	AQE1	653	6	0.991	554	10	0.982
Forecast ± 2σ	AQE2	653	5	0.992	556	8	0.986
	AQE3	552	106	0.839	556	8	0.986

MONTHLY CO

Method	AQEs	2			3		
		TP	FN	Accuracy	TP	FN	Accuracy
(0,1,5) x (12,1,12)							
SD	AQE1	3	656	0.005	3	561	0.005
Forecast ± σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1) x (12,1,12)							
SD	AQE1	4	655	0.006	4	560	0.007
Forecast ± σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1) x (12,0,12)							
SD	AQE1	2	657	0.003	2	562	0.004
Forecast ± σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1) x (24,1,24)							
SD	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	0	658	0.000	0	564	0.000
(0,1,5)							
SD	AQE1	0	659	0.000	0	564	0.000
Forecast ± σ	AQE2	0	568	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1)							
SD	AQE1	0	659	0.000	0	564	0.000
Forecast ± σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(0,1,5) x (12,1,12)							
SD	AQE1	7	652	0.011	6	558	0.011
Forecast ± 2σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1) x (12,1,12)							
SD	AQE1	6	653	0.009	6	558	0.011
Forecast ± 2σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1) x (12,0,12)							
SD	AQE1	6	653	0.009	6	558	0.011
Forecast ± 2σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000

(1,1,1) x (24,1,24)							
SD	AQE1	0	658	0.000	0	564	0.000
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0,1,5)							
SD	AQE1	614	45	0.932	530	34	0.940
Forecast ± 2σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	658	0.000	0	564	0.000
(1,1,1)							
SD	AQE1	1	658	0.002	1	563	0.002
Forecast ± 2σ	AQE2	0	658	0.000	0	564	0.000
	AQE3	0	568	0.000	0	564	0.000
(0,1,5) x (12,1,12)							
SD	AQE1	381	278	0.578	414	150	0.734
Forecast ± σ	AQE2	655	3	0.995	562	2	0.996
	AQE3	226	432	0.343	244	320	0.433
(1,1,1) x (12,1,12)							
SD	AQE1	376	283	0.571	406	158	0.720
Forecast ± σ	AQE2	617	41	0.938	555	9	0.984
	AQE3	464	19	0.961	463	101	0.821
(1,1,1) x (12,0,12)							
SD	AQE1	647	12	0.982	554	10	0.982
Forecast ± σ	AQE2	654	4	0.994	562	2	0.996
	AQE3	653	5	0.992	564	0	1.000
(1,1,1) x (24,1,24)							
SD	AQE1	623	36	0.945	529	35	0.938
Forecast ± σ	AQE2	591	67	0.898	555	9	0.984
	AQE3	262	396	0.398	273	291	0.484
(0,1,5)							
SD	AQE1	620	39	0.941	535	29	0.949
Forecast ± σ	AQE2	643	15	0.977	556	8	0.986
	AQE3	650	8	0.988	563	1	0.998
(1,1,1)							
SD	AQE1	628	31	0.953	544	20	0.965
Forecast ± σ	AQE2	655	3	0.995	563	1	0.998
	AQE3	655	3	0.995	564	0	1.000
(0,1,5) x (12,1,12)							
SD	AQE1	647	12	0.982	548	16	0.972
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	562	96	0.854	559	5	0.991

(1,1,1) x (12,1,12)							
SD	AQE1	645	14	0.979	547	17	0.970
Forecast ± 2σ	AQE2	656	2	0.997	563	1	0.998
	AQE3	656	2	0.997	564	0	1.000
(1,1,1) x (12,0,12)							
SD	AQE1	656	3	0.995	562	2	0.996
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	656	2	0.997	564	0	1.000
(1,1,1) x (24,1,24)							
SD	AQE1	655	4	0.994	556	8	0.986
Forecast ± 2σ	AQE2	656	2	0.997	563	1	0.998
	AQE3	503	155	0.764	503	61	0.892
(0,1,5)							
SD	AQE1	654	5	0.992	557	7	0.988
Forecast ± 2σ	AQE2	655	3	0.995	562	2	0.996
	AQE3	656	2	0.997	564	0	1.000
(1,1,1)							
SD	AQE1	655	4	0.994	558	6	0.989
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,5) x (12,1,12)							
DD	AQE1	2	513	0.004	492	72	0.872
Forecast ± σ	AQE2	0	658	0.000	558	6	0.989
	AQE3	0	658	0.000	502	62	0.890
(1,1,1) x (12,1,12)							
DD	AQE1	23	512	0.043	493	71	0.874
Forecast ± σ	AQE2	0	658	0.000	554	10	0.982
	AQE3	0	658	0.000	529	35	0.938
(1,1,1) x (12,0,12)							
DD	AQE1	0	515	0.000	-	-	-
Forecast ± σ	AQE2	0	658	0.000	558	6	0.989
	AQE3	0	658	0.000	554	10	0.982
(1,1,1) x (24,1,24)							
DD	AQE1	20	495	0.039	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	0	658	0.000	551	13	0.977
(0,1,5)							
Dynamic.	AQE1	0	659	0.000	519	45	0.920
Forecast ± σ	AQE2	0	658	0.000	558	6	0.989
	AQE3	0	658	0.000	554	10	0.982

(1,1,1)							
DD	AQE1	0	659	0.000	519	45	0.920
Forecast ± σ	AQE2	0	658	0.000	558	6	0.989
	AQE3	0	658	0.000	555	9	0.984
(0,1,5) x (12,1,12)							
DD	AQE1	5	510	0.010	542	22	0.961
Forecast ± 2σ	AQE2	0	658	0.000	561	3	0.995
	AQE3	0	658	0.000	557	7	0.988
(1,1,1) x (12,1,12)							
DD	AQE1	5	510	0.010	543	21	0.963
Forecast ± 2σ	AQE2	0	658	0.000	560	4	0.993
	AQE3	0	658	0.000	558	6	0.989
(1,1,1) x (12,0,12)							
DD	AQE1	4	511	0.008	-	-	-
Forecast ± 2σ	AQE2	0	658	0.000	561	3	0.995
	AQE3	0	658	0.000	562	2	0.996
(1,1,1) x (24,1,24)							
DD	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	0	658	0.000	562	2	0.996
(0,1,5)							
DD	AQE1	614	45	0.932	547	17	0.970
Forecast ± 2σ	AQE2	0	658	0.000	561	3	0.995
	AQE3	0	658	0.000	562	2	0.996
(1,1,1)							
DD	AQE1	1	658	0.002	547	17	0.970
Forecast ± 2σ	AQE2	0	658	0.000	561	3	0.995
	AQE3	0	658	0.000	562	2	0.996
(0,1,5) x (12,1,12)							
DDCN	AQE1	533	126	0.809	502	62	0.890
Forecast ± σ	AQE2	655	3	0.995	559	5	0.991
	AQE3	647	11	0.983	555	9	0.984
(1,1,1) x (12,1,12)							
DDCN	AQE1	541	118	0.821	434	130	0.770
Forecast ± σ	AQE2	654	4	0.994	559	5	0.991
	AQE3	649	9	0.986	555	9	0.984
(1,1,1) x (12,0,12)							
DDCN	AQE1	647	12	0.982	519	45	0.920
Forecast ± σ	AQE2	654	4	0.994	559	5	0.991
	AQE3	653	5	0.992	556	8	0.986

(1,1,1) x (24,1,24)							
DDCN	AQE1	623	36	0.945	475	89	0.842
Forecast ± σ	AQE2	654	4	0.994	559	5	0.991
	AQE3	647	11	0.983	547	17	0.970
(0,1,5)							
DDCN	AQE1	447	212	0.678	519	45	0.920
Forecast ± σ	AQE2	654	4	0.994	559	5	0.991
	AQE3	645	13	0.980	556	8	0.986
(1,1,1)							
DDCN	AQE1	447	212	0.678	519	45	0.920
Forecast ± σ	AQE2	654	4	0.994	559	5	0.991
	AQE3	645	13	0.980	556	8	0.986
(0,1,5) x (12,1,12)							
DDCN	AQE1	647	12	0.982	545	19	0.966
Forecast ± 2σ	AQE2	657	1	0.998	564	0	1.000
	AQE3	655	3	0.995	563	1	0.998
(1,1,1) x (12,1,12)							
DDCN	AQE1	645	14	0.979	512	52	0.908
Forecast ± 2σ	AQE2	657	1	0.998	563	1	0.998
	AQE3	656	2	0.997	563	1	0.998
(1,1,1) x (12,0,12)							
DDCN	AQE1	656	3	0.995	547	17	0.970
Forecast ± 2σ	AQE2	657	1	0.998	564	0	1.000
	AQE3	656	2	0.997	563	1	0.998
(1,1,1) x (24,1,24)							
DDCN	AQE1	655	4	0.994	531	33	0.941
Forecast ± 2σ	AQE2	657	1	0.998	564	0	1.000
	AQE3	657	1	0.998	562	2	0.996
(0,1,5)							
DDCN	AQE1	617	42	0.936	547	17	0.970
Forecast ± 2σ	AQE2	657	1	0.998	564	0	1.000
	AQE3	653	5	0.992	563	1	0.998
(1,1,1)							
DDCN	AQE1	516	143	0.783	547	17	0.970
Forecast ± 2σ	AQE2	567	1	0.998	564	0	1.000
	AQE3	653	5	0.992	563	1	0.998
(0,1,5) x (12,1,12)							
DDCN	AQE1	3	656	0.005	-	-	-
Forecast ± σ	AQE2	0	658	0.000	-	-	-
	AQE3	1	657	0.002	-	-	-

(1,1,1) x (12,1,12)							
DDCAN	AQE1	4	655	0.006	505	59	0.895
Forecast ± σ	AQE2	0	658	0.000	560	4	0.993
	AQE3	1	657	0.002	561	3	0.995
(1,1,1) x (12,0,12)							
DDCAN	AQE1	2	657	0.003	-	-	-
Forecast ± σ	AQE2	0	658	0.000	-	-	-
	AQE3	0	658	0.000	-	-	-
(1,1,1) x (24,1,24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0,1,5)							
DDCAN	AQE1	0	659	0.000	519	45	0.920
Forecast ± σ	AQE2	0	658	0.000	562	2	0.996
	AQE3	0	658	0.000	563	1	0.998
(1,1,1)							
DDCAN	AQE1	0	659	0.000	519	45	0.920
Forecast ± σ	AQE2	0	658	0.000	562	2	0.996
	AQE3	0	658	0.000	563	1	0.998
(0,1,5) x (12,1,12)							
DDCAN	AQE1	7	652	0.011	-	-	-
Forecast ± 2σ	AQE2	2	656	0.003	-	-	-
	AQE3	2	656	0.003	-	-	-
(1,1,1) x (12,1,12)							
DDCAN	AQE1	6	653	0.009	546	18	0.968
Forecast ± 2σ	AQE2	1	657	0.002	564	0	1.000
	AQE3	2	656	0.003	564	0	1.000
(1,1,1) x (12,0,12)							
DDCAN	AQE1	6	653	0.009	-	-	-
Forecast ± 2σ	AQE2	1	657	0.002	-	-	-
	AQE3	1	657	0.002	-	-	-
(1,1,1) x (24,1,24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0,1,5)							
DDCAN	AQE1	614	45	0.932	547	17	0.970
Forecast ± 2σ	AQE2	0	658	0.000	564	0	1.000
	AQE3	0	658	0.000	564	0	1.000

(1,1,1)							
DDCAN	AQE1	1	548	0.002	547	17	0.970
Forecast ± 2σ	AQE2	0	658	0.000	564	0	1.000
	AQE3	0	658	0.000	564	0	1.000
(0,1,5) x (12,1,12)							
DDCAN	AQE1	414	245	0.628	502	62	0.890
Forecast ± σ	AQE2	655	3	0.995	560	4	0.993
	AQE3	255	403	0.388	561	3	0.995
(1,1,1) x (12,1,12)							
DDCAN	AQE1	376	283	0.571	388	176	0.688
Forecast ± σ	AQE2	617	41	0.938	556	8	0.986
	AQE3	464	194	0.705	552	12	0.979
(1,1,1) x (12,0,12)							
DDCAN	AQE1	647	12	0.982	519	45	0.920
Forecast ± σ	AQE2	655	3	0.995	562	2	0.996
	AQE3	653	5	0.992	563	1	0.998
(1,1,1) x (24,1,24)							
DDCAN	AQE1	623	36	0.945	475	89	0.842
Forecast ± σ	AQE2	593	65	0.901	557	7	0.988
	AQE3	539	119	0.819	546	18	0.968
(0,1,5)							
DDCAN	AQE1	620	39	0.941	518	46	0.918
Forecast ± σ	AQE2	647	11	0.983	562	2	0.996
	AQE3	650	8	0.988	563	1	0.998
(1,1,1)							
DDCAN	AQE1	628	31	0.953	521	43	0.924
Forecast ± σ	AQE2	655	3	0.995	562	2	0.996
	AQE3	655	3	0.995	563	1	0.998
(0,1,5) x (12,1,12)							
DDCAN	AQE1	647	12	0.982	545	19	0.966
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	571	87	0.868	564	0	1.000
(1,1,1) x (12,1,12)							
DDCAN	AQE1	654	14	0.979	512	52	0.908
Forecast ± 2σ	AQE2	656	2	0.997	560	4	0.993
	AQE3	656	2	0.997	563	1	0.998
(1,1,1) x (12,0,12)							
DDCAN	AQE1	656	3	0.995	547	17	0.970
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	656	2	0.997	564	0	1.000

(1,1,1) x (24,1,24)							
DDCAN	AQE1	655	4	0.994	531	33	0.941
Forecast ± 2σ	AQE2	656	2	0.997	562	2	0.996
	AQE3	657	1	0.998	563	1	0.998
(0,1,5)							
DDCAN	AQE1	654	5	0.992	546	18	0.968
Forecast ± 2σ	AQE2	655	3	0.995	564	0	1.000
	AQE3	656	2	0.997	564	0	1.000
(1,1,1)							
DDCAN	AQE1	655	4	0.994	549	15	0.973
Forecast ± 2σ	AQE2	656	2	0.997	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000

MONTHLY HUMIDITY

Methods	AQEs	2			3		
		TP	FN	Accuracy	TP	FN	Accuracy
(1,1,1) x (12,1,12)							
SD	AQE1	316	343	0.480	301	263	0.534
Forecast ± σ	AQE2	289	369	0.439	300	264	0.532
	AQE3	312	346	0.474	270	294	0.479
(0,1,1) x (12,1,12)							
SD	AQE1	304	355	0.461	309	255	0.548
Forecast ± σ	AQE2	305	353	0.464	278	286	0.493
	AQE3	299	359	0.454	303	261	0.537
(2,1,3) x (12,1,12)							
SD	AQE1	304	355	0.461	303	261	0.537
Forecast ± σ	AQE2	315	343	0.479	299	265	0.530
	AQE3	317	341	0.482	275	289	0.488
(1,1,1)							
SD	AQE1	233	426	0.354	171	393	0.303
Forecast ± σ	AQE2	183	475	0.278	121	443	0.215
	AQE3	119	539	0.181	71	493	0.126
(0,1,1)							
SD	AQE1	216	443	0.328	157	407	0.278
Forecast ± σ	AQE2	165	493	0.251	117	447	0.207
	AQE3	149	509	0.226	95	469	0.168
(2,1,3)							
SD	AQE1	198	461	0.300	146	418	0.259
Forecast ± σ	AQE2	138	520	0.210	98	466	0.174
	AQE3	157	501	0.239	103	461	0.183
(1,1,1) x (12,1,12)							
SD	AQE1	530	129	0.804	458	106	0.812
Forecast ± 2σ	AQE2	576	82	0.875	477	87	0.846
	AQE3	478	180	0.726	431	151	0.741
(0,1,1) x (12,1,12)							
SD	AQE1	565	94	0.857	473	91	0.839
Forecast ± 2σ	AQE2	469	189	0.713	420	144	0.745
	AQE3	559	99	0.850	461	103	0.817
(2,1,3) x (12,1,12)							
SD	AQE1	543	116	0.824	469	95	0.832
Forecast ± 2σ	AQE2	518	140	0.787	446	118	0.791
	AQE3	478	180	0.726	413	151	0.732

(1,1,1)							
SD	AQE1	363	296	0.551	333	231	0.590
Forecast ± 2σ	AQE2	327	331	0.497	303	261	0.537
	AQE3	288	370	0.438	241	323	0.427
(0,1,1)							
SD	AQE1	357	302	0.542	328	236	0.582
Forecast ± 2σ	AQE2	321	337	0.488	300	264	0.532
	AQE3	310	348	0.471	270	294	0.479
(2,1,3)							
SD	AQE1	348	311	0.528	324	240	0.574
Forecast ± 2σ	AQE2	308	350	0.468	273	291	0.484
	AQE3	318	340	0.483	275	289	0.488
(1,1,1) x (12,1,12)							
Dynamic.	AQE1	161	354	0.313	317	247	0.562
Forecast ± σ	AQE2	215	443	0.327	380	184	0.674
	AQE3	210	448	0.319	369	195	0.654
(0,1,1) x (12,1,12)							
DD	AQE1	25	310	0.075	-	-	-
Forecast ± σ	AQE2	198	460	0.301	378	186	0.670
	AQE3	192	66	0.744	372	192	0.660
(2,1,3) x (12,1,12)							
DD	AQE1	165	350	0.320	323	241	0.573
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	177	481	0.269	-	-	-
(1,1,1)							
DD	AQE1	233	426	0.354	325	239	0.576
Forecast ± σ	AQE2	184	474	0.280	323	241	0.573
	AQE3	119	539	0.181	317	247	0.562
(0,1,1)							
DD	AQE1	216	443	0.328	325	239	0.576
Forecast ± σ	AQE2	166	492	0.252	323	241	0.573
	AQE3	150	508	0.228	317	247	0.562
(2,1,3)							
DD	AQE1	203	456	0.308	387	177	0.686
Forecast ± σ	AQE2	138	520	0.210	323	241	0.573
	AQE3	143	515	0.217	318	246	0.564
(1,1,1) x (12,1,12)							
DD	AQE1	280	235	0.544	531	33	0.941
Forecast ± 2σ	AQE2	368	290	0.559	539	25	0.956
	AQE3	366	292	0.556	546	18	0.968

(0,1,1) x (12,1,12)							
DD	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	349	309	0.530	540	24	0.957
	AQE3	355	303	0.540	548	16	0.972
(2,1,3) x (12,1,12)							
DD	AQE1	283	232	0.550	537	27	0.952
Forecast ± σ	AQE2	-	-	-	335	323	0.509
	AQE3	-	-	-	-	-	-
(1,1,1)							
DD	AQE1	363	296	0.551	489	75	0.867
Forecast ± σ	AQE2	328	330	0.498	477	87	0.846
	AQE3	289	369	0.439	464	100	0.823
(0,1,1)							
DD	AQE1	357	302	0.542	489	75	0.867
Forecast ± σ	AQE2	321	337	0.488	476	88	0.844
	AQE3	310	348	0.471	465	99	0.824
(2,1,3)							
DD	AQE1	349	310	0.530	540	24	0.957
Forecast ± σ	AQE2	308	350	0.468	476	88	0.844
	AQE3	312	346	0.474	464	100	0.823
(1,1,1) x (12,1,12)							
DDCN	AQE1	563	96	0.854	564	0	1.000
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,1) x (12,1,12)							
DDCN	AQE1	558	101	0.847	564	0	1.000
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(2,1,3) x (12,1,12)							
DDCN	AQE1	564	95	0.856	564	0	1.000
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1)							
DDCN	AQE1	549	110	0.833	563	1	0.998
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,1)							
DDCN	AQE1	544	115	0.825	563	1	0.998
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000

(2,1,3)							
DDCN	AQE1	541	118	0.821	564	0	1.000
Forecast ± σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1) x (12,1,12)							
DDCN	AQE1	646	13	0.980	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,1) x (12,1,12)							
DDCN	AQE1	646	13	0.980	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(2,1,3) x (12,1,12)							
DDCN	AQE1	646	13	0.980	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1)							
DDCN	AQE1	644	15	0.977	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,1)							
DDCN	AQE1	644	15	0.977	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(2,1,3)							
DDCN	AQE1	644	15	0.977	564	0	1.000
Forecast ± 2σ	AQE2	644	15	0.977	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1) x (12,1,12)							
DDCAN	AQE1	274	385	0.416	415	149	0.736
Forecast ± σ	AQE2	245	413	0.372	433	131	0.768
	AQE3	260	398	0.395	429	135	0.761
(0,1,1) x (12,1,12)							
DDCAN	AQE1	386	273	0.586	422	142	0.748
Forecast ± σ	AQE2	401	257	0.609	440	124	0.780
	AQE3	406	252	0.617	440	124	0.780
(2,1,3) x (12,1,12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-

(1,1,1)							
DDCAN	AQE1	261	398	0.396	333	231	0.590
Forecast ± σ	AQE2	254	404	0.386	325	239	0.576
	AQE3	266	392	0.404	320	244	0.567
(0,1,1)							
DDCAN	AQE1	261	398	0.396	333	231	0.590
Forecast ± σ	AQE2	244	414	0.371	327	237	0.580
	AQE3	258	400	0.392	319	245	0.566
(2,1,3)							
DDCAN	AQE1	237	422	0.360	460	104	0.816
Forecast ± σ	AQE2	240	418	0.365	454	110	0.805
	AQE3	253	405	0.384	435	129	0.771
(1,1,1) x (12,1,12)							
DDCAN	AQE1	475	184	0.721	553	11	0.980
Forecast ± 2σ	AQE2	455	203	0.691	558	6	0.989
	AQE3	465	193	0.707	558	6	0.989
(0,1,1) x (12,1,12)							
DDCAN	AQE1	557	102	0.845	554	10	0.982
Forecast ± 2σ	AQE2	606	52	0.921	557	7	0.988
	AQE3	604	54	0.918	557	7	0.988
(2,1,3) x (12,1,12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(1,1,1)							
DDCAN	AQE1	425	234	0.645	490	74	0.869
Forecast ± 2σ	AQE2	430	228	0.653	480	84	0.851
	AQE3	426	232	0.647	464	100	0.823
(0,1,1)							
DDCAN	AQE1	422	237	0.640	490	74	0.869
Forecast ± 2σ	AQE2	416	242	0.632	481	83	0.853
	AQE3	411	247	0.625	465	99	0.824
(2,1,3)							
DDCAN	AQE1	403	256	0.612	540	24	0.957
Forecast ± 2σ	AQE2	408	250	0.620	543	21	0.963
	AQE3	406	252	0.617	542	22	0.961
(1,1,1) x (12,1,12)							
DDCAN	AQE1	488	171	0.741	451	113	0.800
Forecast ± σ	AQE2	419	239	0.637	455	109	0.807
	AQE3	461	197	0.701	447	117	0.793

(0,1,1) x (12,1,12)							
DDCAN	AQE1	444	215	0.674	397	167	0.704
Forecast ± σ	AQE2	435	223	0.661	409	155	0.725
	AQE3	430	228	0.653	398	166	0.706
(2,1,3) x (12,1,12)							
DDCAN	AQE1	409	250	0.621	372	192	0.660
Forecast ± σ	AQE2	413	245	0.628	385	179	0.683
	AQE3	416	242	0.632	374	190	0.663
(1,1,1)							
DDCAN	AQE1	261	398	0.396	333	231	0.590
Forecast ± σ	AQE2	254	404	0.386	325	239	0.576
	AQE3	266	392	0.404	320	244	0.567
(0,1,1)							
DDCAN	AQE1	261	398	0.396	333	231	0.590
Forecast ± σ	AQE2	244	414	0.371	327	237	0.580
	AQE3	258	400	0.392	319	245	0.566
(2,1,3)							
DDCAN	AQE1	237	422	0.360	455	109	0.807
Forecast ± σ	AQE2	139	419	0.249	452	112	0.801
	AQE3	252	406	0.383	435	129	0.771
(1,1,1) x (12,1,12)							
DDCAN	AQE1	636	23	0.965	562	2	0.996
Forecast ± 2σ	AQE2	595	63	0.904	563	1	0.998
	AQE3	635	23	0.965	564	0	1.000
(0,1,1) x (12,1,12)							
DDCAN	AQE1	631	28	0.958	556	8	0.986
Forecast ± 2σ	AQE2	626	32	0.951	562	2	0.996
	AQE3	624	34	0.948	560	4	0.993
(2,1,3) x (12,1,12)							
DDCAN	AQE1	409	250	0.621	372	192	0.660
Forecast ± 2σ	AQE2	413	245	0.628	385	179	0.683
	AQE3	416	242	0.632	374	190	0.663
(1,1,1)							
DDCAN	AQE1	425	234	0.645	490	74	0.869
Forecast ± 2σ	AQE2	430	228	0.653	480	84	0.851
	AQE3	426	232	0.647	464	100	0.823
(0,1,1)							
DDCAN	AQE1	422	237	0.640	490	74	0.869
Forecast ± 2σ	AQE2	416	242	0.632	481	83	0.853
	AQE3	411	247	0.625	465	99	0.824

(2,1,3)							
DDCAN	AQE1	406	253	0.616	539	25	0.956
Forecast ± 2σ	AQE2	405	253	0.616	542	22	0.961
	AQE3	404	254	0.614	540	24	0.957

MONTHLY TEMPERATURE

Methods	AQEs	2			3		
		TP	FN	Accuracy	TP	FN	Accuracy
(1,1,1) x (12,1,12)							
SD	AQE1	126	533	0.191	64	500	0.113
Forecast ± σ	AQE2	190	468	0.289	142	422	0.252
	AQE3	209	449	0.318	147	417	0.261
(1,1,1) x (24,1,24)							
SD	AQE1	138	521	0.209	48	516	0.085
Forecast ± σ	AQE2	90	568	0.137	22	542	0.039
	AQE3	137	521	0.208	58	506	0.103
(0,1,1) x (12,1,12)							
SD	AQE1	237	422	0.360	135	429	0.239
Forecast ± σ	AQE2	123	535	0.187	61	503	0.108
	AQE3	193	465	0.293	140	424	0.248
(1,1,1)							
SD	AQE1	241	418	0.366	121	443	0.215
Forecast ± σ	AQE2	236	422	0.359	124	440	0.220
	AQE3	253	405	0.384	151	413	0.268
(0,1,1)							
SD	AQE1	237	422	0.360	119	445	0.211
Forecast ± σ	AQE2	237	421	0.360	128	436	0.227
	AQE3	247	411	0.375	150	414	0.266
(1,1,1) x (12,1,12)							
SD	AQE1	278	381	0.422	202	362	0.358
Forecast ± 2σ	AQE2	419	239	0.637	294	270	0.521
	AQE3	422	236	0.641	295	269	0.523
(1,1,1) x (24,1,24)							
SD	AQE1	276	383	0.419	208	356	0.369
Forecast ± 2σ	AQE2	194	464	0.295	134	430	0.238
	AQE3	267	391	0.406	191	373	0.339
(0,1,1) x (12,1,12)							
SD	AQE1	449	210	0.681	326	238	0.578
Forecast ± 2σ	AQE2	267	391	0.406	205	359	0.363
	AQE3	411	247	0.625	283	281	0.502
(1,1,1)							
SD	AQE1	433	226	0.657	303	261	0.537
Forecast ± 2σ	AQE2	431	227	0.655	314	250	0.557
	AQE3	432	226	0.657	311	253	0.551

(0,1,1)							
SD	AQE1	431	228	0.654	301	263	0.534
Forecast ± 2σ	AQE2	433	225	0.658	317	247	0.562
	AQE3	428	230	0.650	310	254	0.550
(1,1,1) x (12,1,12)							
DD	AQE1	180	335	0.350	347	217	0.615
Forecast ± σ	AQE2	250	408	0.380	301	263	0.534
	AQE3	220	438	0.334	-	-	-
(1,1,1) x (24,1,24)							
DD	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	238	420	0.362	416	148	0.738
(0,1,1) x (12,1,12)							
DD	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	244	414	0.371	296	268	0.525
	AQE3	227	431	0.345	290	274	0.514
(1,1,1)							
DD	AQE1	193	322	0.375	346	218	0.613
Forecast ± σ	AQE2	234	424	0.356	338	226	0.599
	AQE3	252	406	0.383	317	247	0.562
(0,1,1)							
DD	AQE1	237	422	0.360	334	230	0.592
Forecast ± σ	AQE2	238	420	0.362	338	226	0.599
	AQE3	252	406	0.383	317	247	0.562
(1,1,1) x (12,1,12)							
DD	AQE1	321	194	0.623	523	41	0.927
Forecast ± 2σ	AQE2	429	229	0.652	524	40	0.929
	AQE3	394	264	0.599	-	-	-
(1,1,1) x (24,1,24)							
DD	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	441	2217	0.166	495	69	0.878
(0,1,1) x (12,1,12)							
DD	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	428	230	0.650	525	39	0.931
	AQE3	423	235	0.643	498	66	0.883
(1,1,1)							
DD	AQE1	354	161	0.687	532	32	0.943
Forecast ± 2σ	AQE2	429	229	0.652	460	104	0.816
	AQE3	432	226	0.657	442	122	0.784

(0,1,1)							
DD	AQE1	431	228	0.654	481	83	0.853
Forecast ± 2σ	AQE2	433	225	0.658	460	104	0.816
	AQE3	432	226	0.657	442	122	0.784
(1,1,1) x (12,1,12)							
DDCN	AQE1	554	105	0.841	555	9	0.984
Forecast ± σ	AQE2	650	8	0.988	563	1	0.998
	AQE3	649	9	0.986	559	5	0.991
(1,1,1) x (24,1,24)							
DDCN	AQE1	567	92	0.860	554	10	0.982
Forecast ± σ	AQE2	650	8	0.988	563	1	0.998
	AQE3	649	9	0.986	55	6	0.902
(0,1,1) x (12,1,12)							
DDCN	AQE1	563	96	0.854	555	9	0.984
Forecast ± σ	AQE2	650	8	0.988	563	1	0.998
	AQE3	649	9	0.986	558	6	0.989
(1,1,1)							
DDCN	AQE1	565	94	0.857	552	12	0.979
Forecast ± σ	AQE2	650	8	0.988	563	1	0.998
	AQE3	649	9	0.986	558	6	0.989
(0,1,1)							
DDCN	AQE1	564	95	0.856	552	12	0.979
Forecast ± σ	AQE2	650	8	0.988	563	1	0.998
	AQE3	649	9	0.986	558	6	0.989
(1,1,1) x (12,1,12)							
DDCN	AQE1	632	27	0.959	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1) x (24,1,24)							
DDCN	AQE1	634	25	0.962	563	1	0.998
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(0,1,1) x (12,1,12)							
DDCN	AQE1	636	23	0.965	564	0	1.000
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	568	0	1.000	564	0	1.000
(1,1,1)							
DDCN	AQE1	636	23	0.965	563	1	0.998
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000

(0,1,1)							
DDCN	AQE1	636	23	0.965	563	1	0.998
Forecast ± 2σ	AQE2	658	0	1.000	564	0	1.000
	AQE3	658	0	1.000	564	0	1.000
(1,1,1) x (12,1,12)							
DDCAN	AQE1	420	239	0.637	-	-	-
Forecast ± σ	AQE2	358	300	0.544	-	-	-
	AQE3	376	282	0.571	-	-	-
(1,1,1) x (24,1,24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0,1,1) x (12,1,12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(1,1,1)							
DDCAN	AQE1	397	262	0.602	377	187	0.668
Forecast ± σ	AQE2	359	299	0.546	369	195	0.654
	AQE3	355	303	0.540	360	204	0.638
(0,1,1)							
DDCAN	AQE1	395	264	0.599	377	187	0.668
Forecast ± σ	AQE2	359	299	0.546	369	195	0.654
	AQE3	360	298	0.547	360	204	0.638
(1,1,1) x (12,1,12)							
DDCAN	AQE1	560	99	0.850	-	-	-
Forecast ± 2σ	AQE2	532	126	0.809	-	-	-
	AQE3	536	122	0.815	-	-	-
(1,1,1) x (24,1,24)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(0,1,1) x (12,1,12)							
DDCAN	AQE1	-	-	-	-	-	-
Forecast ± 2σ	AQE2	-	-	-	-	-	-
	AQE3	-	-	-	-	-	-
(1,1,1)							
DDCAN	AQE1	561	98	0.851	501	63	0.888
Forecast ± 2σ	AQE2	552	106	0.839	492	72	0.872
	AQE3	543	115	0.825	487	77	0.863

(0,1,1)							
DDCAN	AQE1	561	98	0.851	501	63	0.888
Forecast ± 2σ	AQE2	551	107	0.837	492	72	0.872
	AQE3	541	117	0.822	487	77	0.863
(1,1,1) x (12,1,12)							
DDCAN	AQE1	368	291	0.558	396	168	0.702
Forecast ± σ	AQE2	325	333	0.494	377	187	0.668
	AQE3	320	338	0.486	386	178	0.684
(1,1,1) x (24,1,24)							
DDCAN	AQE1	391	268	0.593	494	70	0.876
Forecast ± σ	AQE2	447	211	0.679	492	72	0.872
	AQE3	420	238	0.638	488	76	0.865
(0,1,1) x (12,1,12)							
DDCAN	AQE1	344	315	0.522	311	253	0.551
Forecast ± σ	AQE2	386	272	0.587	336	228	0.596
	AQE3	375	283	0.570	336	228	0.596
(1,1,1)							
DDCAN	AQE1	398	261	0.604	377	187	0.668
Forecast ± σ	AQE2	359	299	0.546	369	195	0.654
	AQE3	355	303	0.540	360	204	0.638
(0,1,1)							
DDCAN	AQE1	395	264	0.599	377	187	0.668
Forecast ± σ	AQE2	357	301	0.543	369	195	0.654
	AQE3	358	300	0.544	360	204	0.638
(1,1,1) x (12,1,12)							
DDCAN	AQE1	555	104	0.842	545	19	0.966
Forecast ± 2σ	AQE2	533	125	0.810	549	15	0.973
	AQE3	525	133	0.798	551	13	0.977
(1,1,1) x (24,1,24)							
DDCAN	AQE1	555	104	0.842	538	26	0.954
Forecast ± 2σ	AQE2	578	80	0.878	540	24	0.957
	AQE3	567	91	0.862	540	24	0.957
(0,1,1) x (12,1,12)							
DDCAN	AQE1	548	111	0.832	542	22	0.961
Forecast ± 2σ	AQE2	550	108	0.836	547	17	0.970
	AQE3	538	120	0.818	550	14	0.975
(1,1,1)							
DDCAN	AQE1	561	98	0.851	501	63	0.888
Forecast ± 2σ	AQE2	552	106	0.839	92	72	0.561
	AQE3	543	115	0.825	487	77	0.863

(0,1,1)							
DDCAN	AQE1	561	98	0.851	501	63	0.888
Forecast ± 2σ	AQE2	551	107	0.837	492	72	0.872
	AQE3	541	117	0.822	487	77	0.863

WEEKLY NO2

METHODS	WK1-WK2			WK2-WK3			WK3-WK4			WK4-WK5			WK5-WK6			WK6-WK7			WK7-WK8			WK8-WK9			WK9-WK10			WK10-WK11			WK11-WK12		
	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy
DD. (0,1,5). Forecast ± σ																																	
AQE1	145	22	0.868	106	62	0.631	162	6	0.964	146	8	0.948	165	3	0.982	145	23	0.863	24	1	0.960	144	24	0.857	164	4	0.976	126	10	0.9265	90	2	0.978
AQE2	126	41	0.754	107	61	0.637	149	19	0.887	113	41	0.734	103	65	0.613	131	37	0.780	91	77	0.542	135	33	0.804	116	52	0.690	77	59	0.5662	22	70	0.239
AQE3	131	36	0.784	120	48	0.714	134	34	0.798	91	63	0.591	85	83	0.506	84	84	0.500	140	28	0.833	101	67	0.601	71	97	0.423	87	49	0.6397	49	43	0.533
DD. (1,1,1). Forecast ± σ																																	
AQE1	147	20	0.880	104	64	0.619	161	7	0.958	146	8	0.948	165	3	0.982	144	24	0.857	24	1	0.960	144	24	0.857	164	4	0.976	126	10	0.9265	90	2	0.978
AQE2	135	32	0.808	98	70	0.583	153	15	0.911	112	42	0.727	103	65	0.613	143	25	0.851	87	81	0.518	132	36	0.786	113	55	0.673	79	57	0.5809	21	71	0.228
AQE3	119	48	0.713	100	68	0.595	166	2	0.988	91	63	0.591	61	107	0.363	84	84	0.500	141	27	0.839	89	79	0.530	71	97	0.423	85	51	0.6250	49	43	0.533
DD. (0,1,5). Forecast ± 2σ																																	
AQE1	164	3	0.982	137	31	0.815	167	1	0.994	148	6	0.961	167	1	0.994	163	5	0.970	168	1	0.994	161	7	0.958	167	1	0.994	135	1	0.9926	92	0	1.000
AQE2	166	1	0.994	165	3	0.982	168	0	1.000	152	2	0.987	165	3	0.982	167	1	0.994	161	7	0.958	165	3	0.982	156	12	0.929	131	5	0.9632	75	17	0.815
AQE3	165	2	0.988	167	1	0.994	168	0	1.000	142	12	0.922	121	47	0.720	166	2	0.988	167	1	0.994	162	6	0.964	10	18	0.357	131	5	0.9632	86	6	0.935
DD. (1,1,1). Forecast ± 2σ																																	
AQE1	164	3	0.982	136	32	0.810	167	1	0.994	148	6	0.961	167	1	0.994	163	5	0.970	168	1	0.994	161	7	0.958	167	1	0.994	135	1	0.9926	92	0	1.000
AQE2	167	0	1.000	160	8	0.952	168	0	1.000	152	2	0.987	165	3	0.982	167	1	0.994	159	9	0.946	165	3	0.982	154	14	0.917	132	4	0.9706	75	17	0.815
AQE3	163	4	0.976	155	13	0.923	168	0	1.000	142	12	0.922	167	1	0.994	16	2	0.889	167	1	0.994	161	7	0.958	150	18	0.893	132	4	0.9706	86	6	0.935
DDCN. (0,1,5). Forecast ± σ.																																	
AQE1	146	21	0.874	118	50	0.702	163	5	0.970	148	6	0.961	165	3	0.982	148	20	0.881	166	3	0.982	145	23	0.863	168	0	1.000	127	9	0.9338	90	2	0.978
AQE2	126	41	0.754	108	60	0.643	154	14	0.917	127	27	0.825	134	34	0.798	149	19	0.887	95	73	0.565	138	30	0.821	142	26	0.845	111	25	0.8162	24	68	0.261

AQE3	131	36	0.784	131	37	0.780	135	33	0.804	108	46	0.701	141	27	0.839	115	53	0.685	146	22	0.869	105	63	0.625	117	51	0.696	118	18	0.8676	58	34	0.630
DDCN. (1,1,1). Forecast ± σ.																																
AQE1	147	20	0.880	116	52	0.690	162	6	0.964	148	6	0.961	165	3	0.982	147	21	0.875	166	3	0.982	145	23	0.863	168	0	1.000	127	9	0.934	90	2	0.978
AQE2	135	32	0.808	99	69	0.589	155	13	0.923	127	27	0.825	134	34	0.798	152	16	0.905	91	77	0.542	135	33	0.804	138	30	0.821	113	23	0.831	23	69	0.250
AQE3	119	48	0.713	104	64	0.619	167	1	0.994	108	46	0.701	139	29	0.827	115	53	0.685	146	22	0.869	94	74	0.560	117	51	0.696	119	17	0.875	58	34	0.630
DDCN. (0,1,5). Forecast ± 2σ.																																
AQE1	165	2	0.988	164	4	0.976	168	0	1.000	153	1	0.994	168	0	1.000	165	3	0.982	169	0	1.000	161	7	0.958	168	0	1.000	135	1	0.993	92	0	1.000
AQE2	166	1	0.994	166	2	0.988	168	0	1.000	152	2	0.987	166	2	0.988	168	0	1.000	161	7	0.958	165	3	0.982	168	0	1.000	132	4	0.971	77	15	0.837
AQE3	165	2	0.988	167	1	0.994	168	0	1.000	153	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	164	4	0.976	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (1,1,1). Forecast ± 2σ.																																
AQE1	165	2	0.988	164	4	0.976	168	0	1.000	153	1	0.994	168	0	1.000	165	3	0.982	169	0	1.000	161	7	0.958	168	0	1.000	135	1	0.993	92	0	1.000
AQE2	167	0	1.000	161	7	0.958	168	0	1.000	152	2	0.987	166	2	0.988	168	0	1.000	159	9	0.946	165	3	0.982	168	0	1.000	133	3	0.978	77	15	0.837
AQE3	163	4	0.976	159	9	0.946	168	0	1.000	153	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	164	4	0.976	168	0	1.000	136	0	1.000	92	0	1.000
DDCAN. (0,1,5). Forecast ± σ.																																
AQE1	167	0	1.000	166	2	0.988	162	6	0.964	154	0	1.000	168	0	1.000	145	23	0.863	169	0	1.000	167	1	0.994	164	4	0.976	136	0	1.000	92	0	1.000
AQE2	126	41	0.754	108	60	0.643	154	14	0.917	130	24	0.844	114	54	0.679	152	16	0.905	94	74	0.560	140	28	0.833	140	28	0.833	111	25	0.816	22	70	0.239
AQE3	162	5	0.970	134	34	0.798	134	34	0.798	154	0	1.000	167	1	0.994	89	79	0.530	163	5	0.970	167	1	0.994	95	73	0.565	136	0	1.000	50	42	0.543
DDCAN. (1,1,1). Forecast ± σ.																																
AQE1	164	3	0.982	160	8	0.952	167	1	0.994	154	0	1.000	168	0	1.000	144	24	0.857	169	0	1.000	168	0	1.000	164	4	0.976	136	0	1.000	92	0	1.000
AQE2	135	35	0.794	99	69	0.589	153	15	0.911	130	24	0.844	114	54	0.679	144	24	0.857	89	79	0.530	136	32	0.810	140	28	0.833	113	23	0.831	21	71	0.228
AQE3	129	38	0.772	116	52	0.690	166	2	0.988	154	0	1.000	165	3	0.982	87	81	0.518	163	5	0.970	167	1	0.994	96	72	0.571	136	0	1.000	50	42	0.543
DDCAN. (0,1,5). Forecast ± 2σ.																																
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	164	4	0.976	169	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	166	1	0.994	166	2	0.988	168	0	1.000	152	2	0.987	165	3	0.982	168	0	1.000	161	7	0.958	165	3	0.982	168	0	1.000	131	5	0.963	75	17	0.815

AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	87	5	0.946
DDCAN. (1,1,1). Forecast ± 2σ.																																	
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	164	4	0.976	169	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	161	7	0.958	168	0	1.000	152	2	0.987	165	3	0.982	168	0	1.000	159	9	0.946	165	3	0.982	167	1	0.994	132	4	0.971	75	17	0.815
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	87	5	0.946

WEEKLY CO

Methods	WK1-WK2			WK2-WK3			WK3-WK4			WK4-WK5			WK5-WK6			WK6-WK7			WK7-WK8			WK8-WK9			WK9-WK10			WK10-WK11			WK11-WK12		
	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy
DD. (0,1,5). Forecast ± σ																																	
AQE1	164	3	0.982	162	6	0.964	142	26	0.845	145	9	0.942	157	11	0.935	152	16	0.905	24	1	0.960	165	3	0.982	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	163	4	0.976	165	3	0.982	164	4	0.976	1	153	0.006	168	0	1.000	164	4	0.976	167	1	0.994	166	2	0.988	167	1	0.994	134	2	0.985	91	1	0.989
AQE3	167	0	1.000	166	2	0.988	163	5	0.970	2	152	0.013	167	1	0.994	164	4	0.976	166	2	0.988	164	4	0.976	167	1	0.994	131	5	0.963	90	2	0.978
DD. (1,1,1). Forecast ± σ																																	
AQE1	164	3	0.982	135	33	0.804	142	26	0.845	0	154	0.000	158	10	0.940	157	11	0.935	24	1	0.960	165	3	0.982	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	167	0	1.000	165	3	0.982	164	4	0.976	0	154	0.000	168	0	1.000	164	4	0.976	167	1	0.994	166	2	0.988	167	1	0.994	134	2	0.985	91	1	0.989
AQE3	167	0	1.000	162	6	0.964	163	5	0.970	0	154	0.000	167	1	0.994	164	4	0.976	166	2	0.988	164	4	0.976	167	1	0.994	131	5	0.963	89	3	0.967
DD. (0,1,5). Forecast ± 2σ																																	
AQE1	167	0	1.000	167	1	0.994	152	16	0.905	151	3	0.981	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	167	1	0.994	166	2	0.988	2	152	0.013	168	0	1.000	166	2	0.988	168	0	1.000	167	1	0.994	168	0	1.000	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	163	5	0.970	6	148	0.039	168	0	1.000	167	1	0.994	168	0	1.000	167	1	0.994	168	0	1.000	134	2	0.985	91	1	0.989
DD. (1,1,1). Forecast ± 2σ																																	
AQE1	167	0	1.000	163	5	0.970	152	16	0.905	4	150	0.026	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	167	1	0.994	166	2	0.988	1	153	0.006	168	0	1.000	166	2	0.988	168	0	1.000	167	1	0.994	168	0	1.000	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	163	5	0.970	0	154	0.000	168	0	1.000	167	1	0.994	168	0	1.000	167	1	0.994	168	0	1.000	134	2	0.985	91	1	0.989
DDCN. (0,1,5). Forecast ± σ.																																	
AQE1	164	3	0.982	162	6	0.964	142	26	0.845	145	9	0.942	157	11	0.935	152	16	0.905	155	14	0.917	164	4	0.976	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	163	4	0.976	166	2	0.988	164	4	0.976	147	7	0.955	168	0	1.000	166	2	0.988	167	1	0.994	167	1	0.994	167	1	0.994	135	1	0.993	91	1	0.989
AQE3	167	0	1.000	166	2	0.988	163	5	0.970	145	9	0.942	167	1	0.994	166	2	0.988	167	1	0.994	166	2	0.988	167	1	0.994	132	4	0.971	90	2	0.978

DDCN. (1,1,1). Forecast ± σ.																															
AQE1	164	3	0.982	135	33	0.804	142	26	0.845	120	34	0.779	158	10	0.940	157	11	0.935	155	14	0.917	164	4	0.976	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	167	0	1.000	166	2	0.988	164	4	0.976	146	8	0.948	168	0	1.000	166	2	0.988	167	1	0.994	167	1	0.994	167	1	0.994	135	1	0.993	91	1	0.989
AQE3	167	0	1.000	164	4	0.976	163	5	0.970	143	11	0.929	167	1	0.994	166	2	0.988	167	1	0.994	166	2	0.988	167	1	0.994	132	4	0.971	89	3	0.967
DDCN. (0,1,5). Forecast ± 2σ.																															
AQE1	167	0	1.000	167	1	0.994	152	16	0.905	151	3	0.981	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	68	0	1.000	167	1	0.994	149	5	0.968	168	0	1.000	167	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	163	5	0.970	153	1	0.994	168	0	1.000	167	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	134	2	0.985	91	1	0.989
DDCN. (1,1,1). Forecast ± 2σ.																															
AQE1	167	0	1.000	163	5	0.970	152	16	0.905	134	20	0.870	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	168	0	1.000	167	1	0.994	149	5	0.968	168	0	1.000	167	1	0.994	1680		1.000	168	0	1.000	168	0	1.000	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	163	5	0.970	151	3	0.981	168	0	1.000	167	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	134	2	0.985	91	1	0.989
DDCAN. (0,1,5). Forecast ± σ.																															
AQE1	164	3	0.982	162	6	0.964	142	26	0.845	148	6	0.961	157	11	0.935	152	16	0.905	155	14	0.917	164	4	0.976	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	166	1	0.994	168	0	1.000	167	1	0.994	149	5	0.968	168	0	1.000	167	1	0.994	168	0	1.000	167	1	0.994	167	1	0.994	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	164	4	0.976	149	5	0.968	168	0	1.000	167	1	0.994	168	0	1.000	167	1	0.994	167	1	0.994	136	0	1.000	92	0	1.000
DDCN. (1,1,1). Forecast ± σ.																															
AQE1	164	3	0.982	135	33	0.804	142	26	0.845	3	151	0.019	158	10	0.940	157	11	0.935	155	14	0.917	164	4	0.976	160	8	0.952	115	21	0.846	69	23	0.750
AQE2	167	0	1.000	165	3	0.982	167	1	0.994	0	154	0.000	168	0	1.000	168	0	1.000	168	0	1.000	167	1	0.994	167	1	0.994	135	1	0.993	92	0	1.000
AQE3	167	0	1.000	163	5	0.970	164	4	0.976	0	154	0.000	168	0	1.000	167	1	0.994	168	0	1.000	167	1	0.994	167	1	0.994	136	0	1.000	92	0	1.000
DDCN. (0,1,5). Forecast ± 2σ.																															
AQE1	167	0	1.000	167	1	0.994	152	16	0.905	151	3	0.981	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	151	3	0.981	168	0	1.000	168	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	167	1	0.994	152	2	0.987	168	0	1.000	168	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	136	0	1.000	92	0	1.000

DDCN. (1,1,1). Forecast ± 2σ.																																
AQE1	167	0	1.000	163	5	0.970	152	16	0.905	8	146	0.052	167	1	0.994	167	1	0.994	167	2	0.988	166	2	0.988	164	4	0.976	133	3	0.978	82	10	0.891
AQE2	167	0	1.000	167	1	0.994	168	0	1.000	1	153	0.006	168	0	1.000	168	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	167	1	0.994	1	153	0.006	168	0	1.000	168	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	136	0	1.000	92	0	1.000

WEEKLY HUMIDITY

Methods	WK1-WK2			WK2-WK3			WK3-WK4			WK4-WK5			WK5-WK6			WK6-WK7			WK7-WK8			WK8-WK9			WK9-WK10			WK10-WK11			WK11-WK12		
	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy
DD. (1,1,1) Forecast ± σ																																	
AQE1	120	47	0.719	99	69	0.589	87	81	0.518	77	77	0.500	98	70	0.583	105	63	0.625	14	11	0.560	5	163	0.030	86	82	0.512	56	80	0.412	24	68	0.261
AQE2	126	41	0.754	98	70	0.583	98	70	0.583	71	83	0.461	98	70	0.583	87	81	0.518	94	74	0.560	110	58	0.655	90	78	0.536	62	74	0.456	26	66	0.283
AQE3	130	37	0.778	94	74	0.560	101	67	0.601	69	85	0.448	99	69	0.589	84	84	0.500	90	78	0.536	109	59	0.649	90	78	0.536	56	80	0.412	26	66	0.283
DD. (2,1,3) Forecast ± σ																																	
AQE1	124	43	0.743	94	74	0.560	87	81	0.518	84	70	0.545	86	82	0.512	111	57	0.661	15	10	0.600	136	32	0.810	88	80	0.524	79	57	0.581	-	-	-
AQE2	130	37	0.778	95	73	0.565	92	76	0.548	77	77	0.500	98	70	0.583	92	76	0.548	89	79	0.530	105	63	0.625	92	76	0.548	62	74	0.456	27	65	0.293
AQE3	130	37	0.778	94	74	0.560	96	72	0.571	69	85	0.448	100	68	0.595	85	83	0.506	70	98	0.417	98	70	0.583	91	77	0.542	71	65	0.522	26	66	0.283
DD. (0,1,1) Forecast ± σ																																	
AQE1	124	43	0.743	99	69	0.589	87	81	0.518	77	77	0.500	98	70	0.583	106	62	0.631	14	11	0.560	132	36	0.786	86	82	0.512	56	80	0.412	24	68	0.261
AQE2	132	35	0.790	102	66	0.607	98	70	0.583	70	84	0.455	98	70	0.583	88	80	0.524	94	74	0.560	110	58	0.655	90	78	0.536	63	73	0.463	26	66	0.283
AQE3	135	32	0.808	96	72	0.571	102	66	0.607	69	85	0.448	100	68	0.595	85	83	0.506	90	78	0.536	110	58	0.655	90	78	0.536	56	80	0.412	26	66	0.283
DD. (1,1,1) Forecast ± 2σ																																	
AQE1	158	9	0.946	149	19	0.887	133	35	0.792	134	20	0.870	138	30	0.821	153	15	0.911	149	20	0.882	166	2	0.988	120	48	0.714	95	41	0.699	72	20	0.783
AQE2	158	9	0.946	148	20	0.881	142	26	0.845	117	37	0.760	133	35	0.792	138	30	0.821	145	23	0.863	165	3	0.982	124	44	0.738	97	39	0.713	74	18	0.804
AQE3	159	8	0.952	150	18	0.893	141	27	0.839	120	34	0.779	128	40	0.762	134	34	0.798	136	32	0.810	163	5	0.970	121	47	0.720	96	43	0.691	72	20	0.783
DD. (2,1,3) Forecast ± 2σ																																	
AQE1	159	8	0.952	144	24	0.857	133	35	0.792	137	17	0.890	153	15	0.911	155	13	0.923	153	16	0.905	164	4	0.976	121	47	0.720	121	11	0.917	-	-	-
AQE2	158	9	0.946	138	30	0.821	136	32	0.810	130	24	0.844	130	38	0.774	142	26	0.845	151	17	0.899	162	6	0.964	129	39	0.768	97	39	0.713	74	18	0.804
AQE3	159	8	0.952	140	28	0.833	137	31	0.815	114	40	0.740	123	45	0.732	134	34	0.798	157	11	0.935	159	9	0.946	128	40	0.762	122	14	0.897	72	20	0.783

DD. (0,1,1) Forecast ± 2σ																															
AQE1	159	8	0.952	146	22	0.869	133	35	0.792	134	20	0.870	138	30	0.821	153	15	0.911	148	21	0.876	166	2	0.988	121	47	0.720	96	40	0.706	69	23	0.750
Methods	159	8	0.952	142	26	0.845	142	26	0.845	114	40	0.740	130	38	0.774	141	27	0.839	145	23	0.863	164	4	0.976	124	44	0.738	97	39	0.713	73	19	0.793
AQE3	163	4	0.976	141	27	0.839	142	26	0.845	119	35	0.773	128	40	0.762	134	34	0.798	137	31	0.815	163	5	0.970	121	47	0.720	96	43	0.691	72	20	0.783
DDCN. (1,1,1). Forecast ± σ.																															
AQE1	167	0	1.000	168	0	1.000	163	5	0.970	154	0	1.000	168	0	1.000	168	0	1.000	162	7	0.959	168	0	1.000	167	1	0.994	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (2,1,3). Forecast ± σ.																															
AQE1	167	0	1.000	168	0	1.000	163	5	0.970	154	0	1.000	168	0	1.000	168	0	1.000	162	7	0.959	168	0	1.000	167	1	0.994	136	0	1.000	-	-	-
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (0,1,1). Forecast ± σ.																															
AQE1	167	0	1.000	168	0	1.000	163	5	0.970	154	0	1.000	168	0	1.000	168	0	1.000	162	7	0.959	168	0	1.000	167	1	0.994	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (1,1,1). Forecast ± 2σ.																															
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	166		1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (2,1,3). Forecast ± 2σ.																															
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	167	2	0.988	168	0	1.000	168	0	1.000	136	0	1.000	-	-	-
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000

DDCN. (0,1,1). Forecast ± 2σ.																																
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	166	3	0.982	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCAN. (1,1,1). Forecast ± σ.																																
AQE1	122	45	0.731	133	35	0.792	132	36	0.786	82	72	0.532	100	68	0.595	105	63	0.625	106	63	0.627	111	57	0.661	107	61	0.637	66	70	0.485	24	68	0.261
AQE2	126	41	0.754	122	46	0.726	98	70	0.583	75	79	0.487	105	63	0.625	87	81	0.518	121	47	0.720	110	58	0.655	103	65	0.613	68	68	0.500	26	66	0.283
AQE3	130	37	0.778	111	57	0.661	103	65	0.613	75	79	0.487	106	62	0.631	86	82	0.512	113	55	0.673	109	59	0.649	101	67	0.601	66	70	0.485	26	66	0.283
DDCAN. (2,1,3). Forecast ± σ.																																
AQE1	124	43	0.743	124	44	0.738	124	44	0.738	92	62	0.597	120	48	0.714	111	57	0.661	125	44	0.740	107	61	0.637	110	58	0.655	105	31	0.772	-	-	-
AQE2	130	37	0.778	114	54	0.679	92	76	0.548	78	76	0.506	140	28	0.833	92	76	0.548	138	30	0.821	105	63	0.625	111	57	0.661	103	33	0.757	-	-	-
AQE3	130	37	0.778	108	60	0.643	97	71	0.577	83	71	0.539	132	36	0.786	91	77	0.542	26	42	0.382	98	70	0.583	107	61	0.637	95	41	0.699	-	-	-
DDCAN. (0,1,1). Forecast ± σ.																																
AQE1	137	37	0.787	123	45	0.732	132	36	0.786	82	72	0.532	100	68	0.595	106	62	0.631	108	61	0.639	111	57	0.661	107	61	0.637	68	68	0.500	24	68	0.261
AQE2	132	35	0.790	112	56	0.667	98	70	0.583	57	59	0.491	1405	63	0.957	88	80	0.524	121	47	0.720	110	58	0.655	103	65	0.613	68	68	0.500	26	66	0.283
AQE3	135	32	0.808	109	9	0.924	103	65	0.613	74	80	0.481	106	62	0.631	86	82	0.512	113	55	0.673	110	58	0.655	101	67	0.601	66	70	0.485	26	66	0.283
DDCAN. (1,1,1). Forecast ± 2σ.																																
AQE1	158	9	0.946	158	10	0.940	149	19	0.887	135	19	0.877	148	20	0.881	153	15	0.911	157	12	0.929	166	2	0.988	145	23	0.863	109	27	0.801	72	20	0.783
AQE2	158	9	0.946	154	14	0.917	142	26	0.845	134	20	0.870	154	14	0.917	138	30	0.821	162	6	0.964	165	3	0.982	141	27	0.839	108	28	0.794	74	18	0.804
AQE3	159	8	0.952	152	16	0.905	141	27	0.839	132	22	0.857	155	13	0.923	134	34	0.798	159	9	0.946	163	5	0.970	136	32	0.810	104	32	0.765	72	20	0.783
DDCAN. (2,1,3). Forecast ± 2σ.																																
AQE1	159	8	0.952	154	14	0.917	148	20	0.881	137	17	0.890	158	10	0.940	155	13	0.923	165	4	0.976	164	4	0.976	151	17	0.899	130	6	0.956	-	-	-
AQE2	158	9	0.946	150	18	0.893	137	31	0.815	134	20	0.870	168	0	1.000	142	26	0.845	166	2	0.988	162	6	0.964	144	24	0.857	132	4	0.971	-	-	-
AQE3	159	8	0.952	151	17	0.899	137	31	0.815	140	14	0.909	168	0	1.000	134	34	0.798	167	1	0.994	159	9	0.946	139	29	0.827	130	6	0.956	-	-	-

DDCAN. (0,1,1). Forecast ± 2σ.																															
AQE1	159	8	0.952	156	12	0.929	149	19	0.887	135	19	0.877	148	20	0.881	153	15	0.911	157	12	0.929	166	2	0.988	145	23	0.863	109	27	0.801	69	23	0.750
AQE2	159	8	0.952	152	16	0.905	142	26	0.845	134	20	0.870	154	14	0.917	141	27	0.839	162	6	0.964	164	4	0.976	141	27	0.839	108	28	0.794	73	19	0.793
AQE3	163	4	0.976	151	17	0.899	142	26	0.845	132	22	0.857	155	13	0.923	134	34	0.798	159	9	0.946	163	5	0.970	136	32	0.810	104	32	0.765	72	20	0.783

WEEKLY TEMPERATURE

Methods	WK1-WK2			WK2-WK3			WK3-WK4			WK4-WK5			WK5-WK6			WK6-WK7			WK7-WK8			WK8-WK9			WK9-WK10			WK10-WK11			WK11-WK12		
	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy	TP	FN	Accuracy
DD. (1,1,1) Forecast ± σ																																	
AQE1	145	22	0.868	28	140	0.167	57	111	0.339	102	52	0.662	109	59	0.649	147	21	0.875	11	14	0.440	84	84	0.5	96	72	0.571	60	76	0.441	76	16	0.826
AQE2	141	26	0.844	45	123	0.268	108	60	0.643	45	109	0.292	102	66	0.607	136	32	0.81	31	137	0.185	112	56	0.667	100	68	0.595	65	71	0.478	76	16	0.826
AQE3	148	19	0.886	67	101	0.399	105	63	0.625	44	110	0.286	97	71	0.577	127	41	0.756	23	145	0.137	104	64	0.619	99	69	0.589	60	76	0.441	52	40	0.565
DD. (0,1,1) Forecast ± σ																																	
AQE1	150	17	0.898	29	139	0.173	120	48	0.714	50	104	0.325	109	59	0.649	147	21	0.875	11	14	0.440	101	67	0.601	95	73	0.565	60	76	0.441	68	24	0.739
AQE2	146	21	0.874	44	124	0.262	108	60	0.643	45	109	0.292	102	66	0.607	136	32	0.810	31	137	0.185	112	56	0.667	99	69	0.589	65	71	0.478	68	24	0.739
AQE3	151	16	0.904	64	104	0.381	105	63	0.625	44	110	0.286	99	69	0.589	127	41	0.756	25	143	0.149	106	62	0.631	99	69	0.589	60	76	0.441	52	40	0.565
DD. (1,1,1) Forecast ± 2σ																																	
AQE1	165	2	0.988	93	75	0.554	139	29	0.827	135	19	0.877	158	10	0.940	164	4	0.976	114	55	0.675	158	10	0.940	128	40	0.762	104	32	0.765	90	2	0.978
AQE2	165	2	0.988	97	71	0.577	159	9	0.946	100	54	0.649	158	10	0.940	157	11	0.935	100	68	0.595	157	11	0.935	126	42	0.750	103	33	0.757	91	1	0.989
AQE3	165	2	0.988	108	60	0.643	155	13	0.923	92	62	0.597	153	15	0.911	157	11	0.935	96	72	0.571	154	14	0.917	125	43	0.744	100	36	0.735	92	0	1.000
DD. (0,1,1) Forecast ± 2σ																																	
AQE1	166	1	0.994	96	75	0.561	155	13	0.923	111	43	0.721	158	10	0.940	164	4	0.976	114	55	0.675	158	10	0.940	128	40	0.762	104	32	0.765	90	2	0.978
AQE2	165	2	0.988	97	71	0.577	159	9	0.946	100	54	0.649	158	10	0.940	157	11	0.935	101	67	0.601	157	11	0.935	125	43	0.744	109	33	0.768	92	0	1.000
AQE3	165	2	0.988	106	62	0.631	155	13	0.923	92	62	0.597	153	15	0.911	157	11	0.935	97	71	0.577	154	14	0.917	124	44	0.738	100	36	0.735	92	0	1.000
DDCN. (1,1,1). Forecast ± σ.																																	
AQE1	167	0	1.000	167	1	0.994	166	2	0.988	152	2	0.987	165	3	0.982	167	1	0.994	158	11	0.935	168	0	1.000	165	3	0.982	132	4	0.971	91	1	0.989
AQE2	167	0	1.000	167	1	0.994	167	1	0.994	153	1	0.994	168	0	1.000	168	0	1.000	166	2	0.988	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	166	2	0.988	154	0	1.000	167	1	0.994	166	2	0.988	166	2	0.988	167	1	0.994	166	2	0.988	136	0	1.000	92	0	1.000

DDCN. (0,1,1). Forecast ± σ.																																	
AQE1	167	0	1.000	167	1	0.994	168	0	1.000	152	2	0.987	165	3	0.982	167	1	0.994	158	11	0.935	168	0	1.000	165	3	0.982	132	4	0.971	91	1	0.989
AQE2	167	0	1.000	167	1	0.994	167	1	0.994	153	1	0.994	168	0	1.000	168	0	1.000	166	2	0.988	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	167	1	0.994	166	2	0.988	154	0	1.000	167	1	0.994	166	2	0.988	166	2	0.988	167	1	0.994	166	2	0.988	136	0	1.000	92	0	1.000
DDCN. (,1,1). Forecast ± 2σ.																																	
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	166	3	0.982	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	68	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCN. (0,1,1). Forecast ± 2σ.																																	
AQE1	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	166	3	0.982	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE2	167	0	1.000	168	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
AQE3	167	0	1.000	168	0	1.000	154	0	1.000	168	0	1.000	167	1	0.994	168	0	1.000	168	0	1.000	168	0	1.000	168	0	1.000	136	0	1.000	92	0	1.000
DDCAN. (1,1,1). Forecast ± σ.																																	
AQE1	145	22	0.868	81	87	0.482	124	44	0.738	102	52	0.662	114	54	0.679	147	21	0.875	59	110	0.349	118	50	0.702	120	48	0.714	75	61	0.551	76	16	0.826
AQE2	141	26	0.844	79	89	0.470	136	32	0.810	103	51	0.669	119	49	0.708	136	32	0.810	73	95	0.435	112	56	0.667	115	53	0.685	73	63	0.537	76	16	0.826
AQE3	148	19	0.886	81	87	0.482	125	43	0.744	99	55	0.643	118	50	0.702	129	39	0.768	76	92	0.452	104	64	0.619	111	57	0.661	71	65	0.522	60	32	0.652
DDCAN. (0,1,1). Forecast ± σ.																																	
AQE1	150	17	0.898	77	91	0.458	137	31	0.815	66	88	0.429	14	54	0.206	47	21	0.691	59	110	0.349	117	51	0.696	119	49	0.708	75	61	0.551	68	24	0.739
AQE2	146	21	0.874	76	92	0.452	134	34	0.798	57	97	0.370	119	49	0.708	136	32	0.810	73	95	0.435	112	56	0.667	114	54	0.679	73	63	0.537	68	24	0.739
AQE3	151	16	0.904	81	87	0.482	135	33	0.804	63	91	0.409	118	50	0.702	129	39	0.768	76	92	0.452	106	62	0.631	111	57	0.661	71	65	0.522	52	40	0.565
DDCAN. (1,1,1). Forecast ± 2σ.																																	
AQE1	165	2	0.988	130	38	0.774	164	4	0.976	135	19	0.877	162	6	0.964	164	4	0.976	118	51	0.698	158	10	0.940	152	16	0.905	111	25	0.816	90	2	0.978
AQE2	165	2	0.988	126	42	0.750	163	5	0.970	138	16	0.896	163	5	0.970	157	11	0.935	123	45	0.732	157	11	0.935	147	21	0.875	107	29	0.787	91	1	0.989
AQE3	165	2	0.988	123	45	0.732	163	5	0.970	137	17	0.890	163	5	0.970	157	11	0.935	120	48	0.714	154	14	0.917	146	22	0.869	104	32	0.765	92	0	1.000

DDCAN. (0,1,1). Forecast ± 2σ.																															
AQE1	166	1	0.994	128	40	0.762	164	4	0.976	128	26	0.831	162	6	0.964	164	4	0.976	119	50	0.704	158	10	0.940	152	16	0.905	111	25	0.816	90	2	0.978
AQE2	165	2	0.988	124	44	0.738	160	8	0.952	127	27	0.825	163	5	0.970	157	11	0.935	123	45	0.732	157	11	0.935	147	21	0.875	107	29	0.787	92	0	1.000
AQE3	165	2	0.988	123	45	0.732	161	7	0.958	120	34	0.779	163	5	0.970	157	11	0.935	120	48	0.714	154	14	0.917	146	22	0.869	104	32	0.765	92	0	1.000

Appendix B R Source Code

The following page contains the R codes used in the data training and testing stages.

```
set.seed(777)
##all required libraries are called here
library(dplyr)
library(neuralnet)
library(nnet)
library(nnetpredint)
library(RSNNS)
library(gbm)

## DATASETS
#read AQE files
#AQE1
egg1w1 <- na.omit(read.csv("egg008027a2bc1b0113-week1.csv", stringsAsFactors = FALSE, sep = ","))
egg1w2 <- na.omit(read.csv("egg008027a2bc1b0113-week2.csv", stringsAsFactors = FALSE, sep = ","))
egg1w3 <- na.omit(read.csv("egg008027a2bc1b0113-week3.csv", stringsAsFactors = FALSE, sep = ","))
egg1w4 <- na.omit(read.csv("egg008027a2bc1b0113-week4.csv", stringsAsFactors = FALSE, sep = ","))
egg1w5 <- na.omit(read.csv("egg008027a2bc1b0113-week5.csv", stringsAsFactors = FALSE, sep = ","))
egg1w6 <- na.omit(read.csv("egg008027a2bc1b0113-week6.csv", stringsAsFactors = FALSE, sep = ","))
egg1w7 <- na.omit(read.csv("egg008027a2bc1b0113-week7.csv", stringsAsFactors = FALSE, sep = ","))
egg1w8 <- na.omit(read.csv("egg008027a2bc1b0113-week8.csv", stringsAsFactors = FALSE, sep = ","))
egg1w9 <- na.omit(read.csv("egg008027a2bc1b0113-week9.csv", stringsAsFactors = FALSE, sep = ","))
egg1w10 <- na.omit(read.csv("egg008027a2bc1b0113-week10.csv", stringsAsFactors = FALSE, sep = ","))
egg1w11 <- na.omit(read.csv("egg008027a2bc1b0113-week11.csv", stringsAsFactors = FALSE, sep = ","))
egg1w12 <- na.omit(read.csv("egg008027a2bc1b0113-week12.csv", stringsAsFactors = FALSE, sep = ","))

#AQE2
egg2w1 <- na.omit(read.csv("egg008027a278880113-week1.csv", stringsAsFactors = FALSE, sep = ","))
egg2w2 <- na.omit(read.csv("egg008027a278880113-week2.csv", stringsAsFactors = FALSE, sep = ","))
egg2w3 <- na.omit(read.csv("egg008027a278880113-week3.csv", stringsAsFactors = FALSE, sep = ","))
egg2w4 <- na.omit(read.csv("egg008027a278880113-week4.csv", stringsAsFactors = FALSE, sep = ","))
egg2w5 <- na.omit(read.csv("egg008027a278880113-week5.csv", stringsAsFactors = FALSE, sep = ","))
egg2w6 <- na.omit(read.csv("egg008027a278880113-week6.csv", stringsAsFactors = FALSE, sep = ","))
egg2w7 <- na.omit(read.csv("egg008027a278880113-week7.csv", stringsAsFactors = FALSE, sep = ","))
egg2w8 <- na.omit(read.csv("egg008027a278880113-week8.csv", stringsAsFactors = FALSE, sep = ","))
egg2w9 <- na.omit(read.csv("egg008027a278880113-week9.csv", stringsAsFactors = FALSE, sep = ","))
egg2w10 <- na.omit(read.csv("egg008027a278880113-week10.csv", stringsAsFactors = FALSE, sep = ","))
egg2w11 <- na.omit(read.csv("egg008027a278880113-week11.csv", stringsAsFactors = FALSE, sep = ","))
egg2w12 <- na.omit(read.csv("egg008027a278880113-week12.csv", stringsAsFactors = FALSE, sep = ","))

#AQE3
egg3w1 <- na.omit(read.csv("egg008028735b980112-week1.csv", stringsAsFactors = FALSE, sep = ","))
egg3w2 <- na.omit(read.csv("egg008028735b980112-week2.csv", stringsAsFactors = FALSE, sep = ","))
egg3w3 <- na.omit(read.csv("egg008028735b980112-week3.csv", stringsAsFactors = FALSE, sep = ","))
egg3w4 <- na.omit(read.csv("egg008028735b980112-week4.csv", stringsAsFactors = FALSE, sep = ","))
egg3w5 <- na.omit(read.csv("egg008028735b980112-week5.csv", stringsAsFactors = FALSE, sep = ","))
egg3w6 <- na.omit(read.csv("egg008028735b980112-week6.csv", stringsAsFactors = FALSE, sep = ","))
egg3w7 <- na.omit(read.csv("egg008028735b980112-week7.csv", stringsAsFactors = FALSE, sep = ","))
egg3w8 <- na.omit(read.csv("egg008028735b980112-week8.csv", stringsAsFactors = FALSE, sep = ","))
egg3w9 <- na.omit(read.csv("egg008028735b980112-week9.csv", stringsAsFactors = FALSE, sep = ","))
egg3w10 <- na.omit(read.csv("egg008028735b980112-week10.csv", stringsAsFactors = FALSE, sep = ","))
egg3w11 <- na.omit(read.csv("egg008028735b980112-week11.csv", stringsAsFactors = FALSE, sep = ","))
egg3w12 <- na.omit(read.csv("egg008028735b980112-week12.csv", stringsAsFactors = FALSE, sep = ","))

#ECan
#read Past ECan
airquality <- read.csv("riccartonReformatHOURLYSorted.csv")

#read Current ECan
ecan <- na.omit(read.csv("ECan-3Months.csv", stringsAsFactors = FALSE, sep = ","))
#fix datetime format

egg1Summary <- egg1[,-11] %>% group_by(dfactor) %>% summarise(meantemp=mean(temperature.degC., na.rm=TRUE), meanNO2=mean(no2.ppb., na.rm=TRUE),
            meanCO=mean(co.ppm., na.rm=TRUE), meanHumidity=mean(humidity..., na.rm=TRUE), n())
egg2Summary <- egg2[,-11] %>% group_by(dfactor) %>% summarise(meantemp=mean(temperature.degC., na.rm=TRUE), meanNO2=mean(no2.ppb., na.rm=TRUE),
            meanCO=mean(co.ppm., na.rm=TRUE), meanHumidity=mean(humidity..., na.rm=TRUE), n())
egg3Summary <- egg3[,-11] %>% group_by(dfactor) %>% summarise(meantemp=mean(temperature.degC., na.rm=TRUE), meanNO2=mean(no2.ppb., na.rm=TRUE),
            meanCO=mean(co.ppm., na.rm=TRUE), meanHumidity=mean(humidity..., na.rm=TRUE), n())

ecanconverted <- convertTime(ecan)

allData <- joinAllEgg(egg1Summary, egg2Summary, egg3Summary, ecanconverted, 9, 12, 12, 4)

#manually calculate ARIMA from stats package
```

```
best.aic<-1e8
#maxord <- data.frame(5,5,5)
x.ts <- temperature
n<-length(x.ts)
#for(p in 0:maxord[1]) for(d in 0:maxord[2]) for(q in 0:maxord[3])
for(p in 0:5) for(d in 0:2) for(q in 0:5)
{
 fit<-arima(x.ts,order=c(p,d,q), optim.control = list(maxit = 1000))
 fit.aic<--2*fit$loglik+(log(n)+1)*length(fit$coef)
 if(fit.aic<best.aic)
 {
  best.aic<-fit.aic
  best.fit<-fit
  best.model<-c(p,d,q)
 }
}
list(best.aic,best.fit,best.model)


### FUNCTIONS
## DATA ANALYSIS FUNCTION
#combine 2 data with similar date and time
#dest and source assume have a date field
#REQUIREMENTS: dest needs to be formatted into H:M:S (orignial format only H)
integrateData <- function(source, dest, newVar) {
 source <- ecanw2
 dest <- egg1w2Sorted
 destLength <- length(dest[[1]])

 #create empty new column
 newVar <- c(rep(NA,destLength))

 for(index in 1:destLength)
 {
  position <- match(as.POSIXct(dest$Group.date[index], "%Y/%m/%d %H:%M:%S", tz=""),as.POSIXct(source$date,"%Y/%m/%d %H:%M:%S", tz=""))

  if(!is.na(position)) {
   newVar[index] <- source$Temperature.2m..DegC.[position]
  }
 }

  dest <- cbind(dest,newVar)
}

##join all eggs into columns
##src1 = egg1, src2 = egg2, src3 = egg3, src4 = ecan
##internally, change the variable
##return variables with more rows
joinAllEgg <- function(src1, src2, src3, src4, startmonth, endmonth, startday, endday) {
 newDate <- seq(ISOdatetime(2016, startmonth, startday, 9,0,0, tz=""),ISOdate(2016, endmonth, endday, 23,0,0, tz=""), "hour")
 ldata <- length(newDate)

 #create empty new column
 #CO
 AQE1CO <- c(rep(NA,ldata))
 AQE2CO <- c(rep(NA,ldata))
 AQE3CO <- c(rep(NA,ldata))
 ECanCO <- c(rep(NA,ldata))
 #no2
 AQE1NO2 <- c(rep(NA,ldata))
 AQE2NO2 <- c(rep(NA,ldata))
 AQE3NO2 <- c(rep(NA,ldata))
 ECanNO2 <- c(rep(NA,ldata))
 #temperature
 AQE1Temp <- c(rep(NA,ldata))
 AQE2Temp <- c(rep(NA,ldata))
 AQE3Temp <- c(rep(NA,ldata))
 ECanTemp <- c(rep(NA,ldata))
 #humidity
 AQE1Hum <- c(rep(NA,ldata))
 AQE2Hum <- c(rep(NA,ldata))
 AQE3Hum <- c(rep(NA,ldata))
 ECanHum <- c(rep(NA,ldata))

 for(loop in 1: ldata) {
  #pay attention to different date format across OS
  #position1 <- match(newDate[loop],as.POSIXlt(reconstructDate(src1$dfactor),"%Y-%m-%d %H:%M:%S", tz=""))
  position1 <- match(newDate[loop],as.POSIXct(src1$dfactor,"%Y-%m-%d %H:%M:%S", tz=""))
  #position2 <- match(newDate[loop],as.POSIXlt(reconstructDate(src2$dfactor),"%Y-%m-%d %H:%M:%S", tz=""))
  position2 <- match(newDate[loop],as.POSIXct(src2$dfactor,"%Y-%m-%d %H:%M:%S", tz=""))
  #position3 <- match(newDate[loop],as.POSIXlt(reconstructDate(src3$dfactor),"%Y-%m-%d %H:%M:%S", tz=""))
  position3 <- match(newDate[loop],as.POSIXct(src3$dfactor,"%Y-%m-%d %H:%M:%S", tz=""))
  position4 <- match(newDate[loop],as.POSIXct(src4$date,"%Y-%m-%d %H:%M:%S", tz=""))

  if(!is.na(position1)) {
   #AQE1[loop] <- src1$x[position1]
   AQE1CO[loop] <- src1$meanCO[position1]
```

```
    AQE1NO2[loop] <- src1$meanNO2[position1]
    AQE1Temp[loop] <- src1$meantemp[position1]
    AQE1Hum[loop] <- src1$meanHumidity[position1]
   }
   if(!is.na(position2)) {
    #AQE2[loop] <- src2$x[position2]
    AQE2CO[loop] <- src2$meanCO[position2]
    AQE2NO2[loop] <- src2$meanNO2[position2]
    AQE2Temp[loop] <- src2$meantemp[position2]
    AQE2Hum[loop] <- src2$meanHumidity[position2]
   }
   if(!is.na(position3)) {
    #AQE3[loop] <- src3$x[position3]
    AQE3CO[loop] <- src3$meanCO[position3]
    AQE3NO2[loop] <- src3$meanNO2[position3]
    AQE3Temp[loop] <- src3$meantemp[position3]
    AQE3Hum[loop] <- src3$meanHumidity[position3]
   }
   if(!is.na(position4)) {
    ECanCO[loop] <- src4$CO..mg.m3.[position4]
    ECanNO2[loop] <- src4$NO2..ug.m3.[position4]
    ECanTemp[loop] <- src4$Temperature.2m..DegC.[position4]
    ECanHum[loop] <- src4$Relative.humidity....[position4]
   }
  }
  newData <- data.frame(newDate, AQE1CO, AQE2CO, AQE3CO, ECanCO, AQE1NO2, AQE2NO2, AQE3NO2, ECanNO2, AQE1Temp, AQE2Temp, AQE3Temp, ECanTemp,
                        AQE1Hum, AQE2Hum, AQE3Hum, ECanHum)
  newData
}

#ECan date format is not recognized by R
#convert a.m./p.m. into recognizable R time format
#this function also address the format of 12:00:00 p.m. and 12:00:00 a.m.
#return standard date format in R: %Y-%m-%d
convertTime <- function(x) {
 x$date <- as.POSIXlt(x$DateTime, "%d/%m/%Y %H:%M:%S", tz="")
 lengthDate <- length(x$DateTime)
 for(index in 1:lengthDate) {
   #dTime <- gsub("([0-9]{2}/[0-9]{2}/[0-9]{4}[ ])", "", x$DateTime[index])
   #dTime <- strsplit(dTime, " ")
   dTime <- strsplit(x$DateTime[index], " ")
   if(dTime[[1]][3] == "p.m." && dTime[[1]][2] != "12:00:00") {
     x$date[index] <- x$date[index] + 12*60*60
   }
   else if(dTime[[1]][3] == "a.m." && dTime[[1]][2] == "12:00:00") {
     x$date[index] <- as.POSIXlt(paste0(dTime[[1]][1]," 00:00:00"),"%d/%m/%Y %H:%M:%S", tz="")
   }
 }
 x
}

## Sample of running ANN simulations using GB method
## Running ANN simulations and save to file
## on LINUX
learningRate <- c(0.1,0.5,0.01,0.05,0.001,0.005,0.0001,0.0005,0.00001,0.00005)
repRate <- c(1,10,100,500,1000,5000,10000,50000)
numOfCV <- c(5, 10)
bestIterMethods <- c("cv", "test", "OOB")
trainPct <- c(0.5, 0.75, 0.9)
bagFraction <- c(0.5, 0.75)

lengthLR <- length(learningRate)
lengthRR <- length(repRate)
lengthCV <- length(numOfCV)
lengthIterMethod <- length(bestIterMethods)
lengthTrainPct <- length(trainPct)
lengthBagFrac <- length(bagFraction)
totalCom <- lengthLR * lengthRR * lengthCV * lengthIterMethod * lengthTrainPct * lengthBagFrac

temp_trainrsquared <- c(rep(NA, totalCom))
temp_tesrsquared <- c(rep(NA, totalCom))
temp_rmse <- c(rep(NA, totalCom))
temp_d <- c(rep(NA, totalCom))
temp_LR <- c(rep(NA, totalCom))
temp_RR <- c(rep(NA, totalCom))
temp_CV <- c(rep(NA, totalCom))
temp_trainPct <- c(rep(NA, totalCom))
temp_itermethod <- c(rep(NA, totalCom))
temp_bagFr <- c(rep(NA, totalCom))

indexCount <- 1

for (act in 1:lengthIterMethod)
{
 for (bagF in 1:lengthBagFrac)
 {
  for (h in 1:lengthCV)
```

```
  {
   for (pct in 1:lengthTrainPct)
   {
    for (lr in 1:lengthLR)
    {
     for (rr in 1:lengthRR)
     {
      result <- runningANNwithBoosting(ECanNO2 ~ AQE1NO2 + AQE2NO2 + AQE3NO2, noNAAllDataNO2, c(7:9), 10,  learningRate[lr], repRate[rr], trainPct[pct], 0.5,
               numOfCV[h], bestIterMethods[act], bagFraction[bagF])

      temp_itermethod[indexCount] <- bestIterMethods[act]
      temp_bagFr[indexCount] <- bagFraction[bagF]
      temp_CV[indexCount] <- numOfCV[h]
      temp_trainPct[indexCount] <- trainPct[pct]
      temp_LR[indexCount] <- learningRate[lr]
      temp_RR[indexCount] <- repRate[rr]
      temp_trainrsquared[indexCount] <- round(result$trainRSquare,3)
      temp_tesrsquared[indexCount] <- round(result$tesRsquare,3)
      temp_rmse[indexCount] <- round(result$rmse,3)
      temp_d[indexCount] <- round(result$d,3)

      indexCount <- indexCount + 1
     }
    }
   }
  }
 }
}

allResult <- data.frame(iterationmethod=temp_itermethod,bagFraction=temp_bagFr, CV=temp_CV,trainingPct=temp_trainPct, learningParameter=temp_LR,
              epoch=temp_RR, r2training=temp_trainrsquared, r2tes=temp_tesrsquared,rmse=temp_rmse, d=temp_d, trainpct=temp_trainPct)
write.csv(allResult, "simulationResult3InputsCONormalised.csv")



## OUTLIER MODULE FUNCTIONS
#x is forecast, h is next month's data
flagOutlierCO <- function(x,h)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 for(loop in 1:numLoop)
 {
  if(is.na(h$meanCO[loop]) || is.na(x$mean[loop]))
  {
   next
  }
   if(x$mean[loop] + stdDevCO < h$meanCO[loop] || x$mean[loop] - stdDevCO > h$meanCO[loop])
  {
   countFN <- countFN +1
  }
  if(x$mean[loop] + stdDevCO >= h$meanCO[loop] && x$mean[loop] - stdDevCO <= h$meanCO[loop])
  {
   countTP <- countTP + 1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierCOWithCheckNeighbor <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  if(is.na(h$meanCO[loop]) || is.na(x$mean[loop]))
  {
   next
  }
  if(x$mean[loop] + (2*stdDevCO) < h$meanCO[loop] || x$mean[loop] - (2*stdDevCO) > h$meanCO[loop])
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
   for(loopNeighbor in 1: numNeighbor)
   {
    #the number of rows may be different
    if(is.na(h$meanCO[loop]) || is.na(y[[loopNeighbor]]$meanCO[loop]))
```

```
      {
        next
      }
      if(y[[loopNeighbor]]$meanCO[loop] + (2*stdDevNeighbor[loopNeighbor,1]) <  h$meanCO[loop] || y[[loopNeighbor]]$meanCO[loop] -
                (2*stdDevNeighbor[loopNeighbor,1]) > h$meanCO[loop] )
      {
        flaggedOutlier <- flaggedOutlier + 1
      }
    }
    #only flagged if the value out of the range of all neighbor
    if(flaggedOutlier == numNeighbor)
    {
      countFN <- countFN +1
    } else {
      #flagged as TP
      countTP <- countTP +1
    }
   }
   if(x$mean[loop] + (2*stdDevCO) >= h$meanCO[loop] && x$mean[loop] - (2*stdDevCO) <= h$meanCO[loop])
   {
    countTP <- countTP + 1
   }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbors' arima model
flagOutlierCOWithCheckNeighborUsingOwnModel <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  if(is.na(h$meanCO[loop]) || is.na(x$mean[loop]))
  {
    next
  }
  if(x$mean[loop] + (2*stdDevCO) < h$meanCO[loop] || x$mean[loop] - (2*stdDevCO) > h$meanCO[loop])
  {
    #it's about to be flagged, but first check the neighbor
    #flagged only if the value is over all the neighbor's value
    flaggedOutlier <- 0
    for(loopNeighbor in 1: numNeighbor)
    {
      #the number of rows may be different
      if(is.na(h$meanCO[loop]) || is.na(y[[loopNeighbor]]$mean[loop]))
      {
        next
      }
      if(y[[loopNeighbor]]$mean[loop] + (2*stdDevNeighbor[loopNeighbor,1]) < h$meanCO[loop] || y[[loopNeighbor]]$mean[loop] - (2*stdDevNeighbor[loopNeighbor,1])
                > h$meanCO[loop])
      {
        flaggedOutlier <- flaggedOutlier + 1
      }
    }
    #only flagged if the value out of the range of all neighbor
    if(flaggedOutlier == numNeighbor)
    {
      countFN <- countFN +1
    } else {
      #flagged as TP
      countTP <- countTP +1
    }
   }
   if(x$mean[loop] + (2*stdDevCO) >= h$meanCO[loop] && x$mean[loop] - (2*stdDevCO) <= h$meanCO[loop])
   {
    countTP <- countTP + 1
   }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

flagOutlierHum <- function(x,h)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 for(loop in 1:numLoop)
 {
```

```
  if(is.na(x$mean[loop]) || is.na(h$meanHumidity[loop]))
  {
    next
  }
  if(x$mean[loop] + (2*stdDevHum) < h$meanHumidity[loop] || x$mean[loop] - (2*stdDevHum) > h$meanHumidity[loop])
  {
    countFN <- countFN +1
  }
  if(x$mean[loop] + (2*stdDevHum) >= h$meanHumidity[loop] && x$mean[loop] - (2*stdDevHum) <= h$meanHumidity[loop])
  {
    countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierHumWithCheckNeighbor <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  #the number of rows may be different
  if(is.na(x$mean[loop]) || is.na(h$meanHumidity[loop]))
  {
    next
  }
  if(x$mean[loop] + (2*stdDevHum) < h$meanHumidity[loop] || x$mean[loop] - (2*stdDevHum) > h$meanHumidity[loop])
  {
    #it's about to be flagged, but first check the neighbor
    #flagged only if the value is over all the neighbor's value
    flaggedOutlier <- 0
    for(loopNeighbor in 1: numNeighbor)
    {
      #the number of rows may be different
      if(is.na(h$meanHumidity[loop]) || is.na(y[[loopNeighbor]]$meanHumidity[loop]))
      {
        next
      }

      if(y[[loopNeighbor]]$meanHumidity[loop] + (2*stdDevNeighbor[loopNeighbor,1]) <  h$meanHumidity[loop] || y[[loopNeighbor]]$meanHumidity[loop] -
                (2*stdDevNeighbor[loopNeighbor,1]) > h$meanHumidity[loop] )
      {
        flaggedOutlier <- flaggedOutlier + 1
      }
    }
    #only flagged if the value out of the range of all neighbor
    if(flaggedOutlier == numNeighbor)
    {
      countFN <- countFN +1
    } else {
      #flagged as TP
      countTP <- countTP +1
    }
  }
  if(x$mean[loop] + (2*stdDevHum) >= h$meanHumidity[loop] && x$mean[loop] - (2*stdDevHum) <= h$meanHumidity[loop])
  {
    countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbors' arima model
flagOutlierHumWithCheckNeighborUsingOwnModel <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  #the number of rows may be different
  if(is.na(x$mean[loop]) || is.na(h$meanHumidity[loop]))
```

```
  {
   next
  }
  if(x$mean[loop] + stdDevHum < h$meanHumidity[loop] || x$mean[loop] - stdDevHum > h$meanHumidity[loop])
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
   for(loopNeighbor in 1: numNeighbor)
   {
    #the number of rows may be different
    if(is.na(h$meanHumidity[loop]) || is.na(y[[loopNeighbor]]$mean[loop]))
    {
     next
    }
    if(y[[loopNeighbor]]$mean[loop] + stdDevNeighbor[loopNeighbor,1] < h$meanHumidity[loop] || y[[loopNeighbor]]$mean[loop] - stdDevNeighbor[loopNeighbor,1] >
              h$meanHumidity[loop])
    {
     flaggedOutlier <- flaggedOutlier + 1
    }
   }
   #only flagged if the value out of the range of all neighbor
   if(flaggedOutlier == numNeighbor)
   {
    countFN <- countFN +1
   } else {
    #flagged as TP
    countTP <- countTP +1
   }
  }
  if(x$mean[loop] + stdDevHum >= h$meanHumidity[loop] && x$mean[loop] - stdDevHum <= h$meanHumidity[loop])
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

flagOutlierNO <- function(x,h)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 for(loopNO in 1:numLoop)
 {
  #skip if the length is not the same
  if(is.na(h$meanNO2[loopNO]) || is.na(x$mean[loopNO]))
  {
   next
  }
  if( (x$mean[loopNO] + stdDevNO < h$meanNO2[loopNO]) || (x$mean[loopNO] - stdDevNO > h$meanNO2[loopNO]) )
  {
   countFN <- countFN +1
  }
  if( (x$mean[loopNO] + stdDevNO >= h$meanNO2[loopNO]) && (x$mean[loopNO] - stdDevNO <= h$meanNO2[loopNO]) )
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierNOWithCheckNeighbor <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 numLoop <- length(x$mean)
 for(loopNO in 1:numLoop)
 {
  #skip if the length is not the same
  if(is.na(h$meanNO2[loopNO]) || is.na(x$mean[loopNO]))
  {
   next
  }
  if( (x$mean[loopNO] + (2*stdDevNO) < h$meanNO2[loopNO]) || (x$mean[loopNO] - (2*stdDevNO) > h$meanNO2[loopNO]) )
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
```

```
    for(loopNeighbor in 1: numNeighbor)
    {
      #the number of rows may be different
      if(is.na(h$meanNO2[loopNO]) || is.na(y[[loopNeighbor]]$meanNO2[loopNO]))
      {
        next
      }
      if(y[[loopNeighbor]]$meanNO2[loopNO] + (2*stdDevNeighbor[loopNeighbor,1]) <  h$meanNO2[loopNO] || y[[loopNeighbor]]$meanNO2[loopNO] -
                (2*stdDevNeighbor[loopNeighbor,1]) > h$meanNO2[loopNO] )
      {
        flaggedOutlier <- flaggedOutlier + 1
      }
    }
    #only flagged if the value out of the range of all neighbor
    if(flaggedOutlier == numNeighbor)
    {
      countFN <- countFN +1
    } else {
      #flagged as TP
      countTP <- countTP +1
    }
  }
  if( (x$mean[loopNO] + (2*stdDevNO) >= h$meanNO2[loopNO]) && (x$mean[loopNO] - (2*stdDevNO) <= h$meanNO2[loopNO]) )
  {
    countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierNOWithCheckNeighborUsingOwnModel <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 numLoop <- length(x$mean)
 for(loopNO in 1:numLoop)
 {
  #skip if the length is not the same
  if(is.na(h$meanNO2[loopNO]) || is.na(x$mean[loopNO]))
  {
    next
  }
  if( (x$mean[loopNO] + (2*stdDevNO) < h$meanNO2[loopNO]) || (x$mean[loopNO] - (2*stdDevNO) > h$meanNO2[loopNO]) )
  {
    #it's about to be flagged, but first check the neighbor
    #flagged only if the value is over all the neighbor's value
    flaggedOutlier <- 0
    for(loopNeighbor in 1: numNeighbor)
    {
      #the number of rows may be different
      if(is.na(h$meanNO2[loopNO]) || is.na(y[[loopNeighbor]]$mean[loopNO]))
      {
        next
      }
      if(y[[loopNeighbor]]$mean[loopNO] + (2*stdDevNeighbor[loopNeighbor,1] < h$meanNO2[loopNO]) || y[[loopNeighbor]]$mean[loopNO] -
                (2*stdDevNeighbor[loopNeighbor,1]) > h$meanNO2[loopNO])
      {
        flaggedOutlier <- flaggedOutlier + 1
      }
    }
    #only flagged if the value out of the range of all neighbor
    if(flaggedOutlier == numNeighbor)
    {
      countFN <- countFN +1
    } else {
      #flagged as TP
      countTP <- countTP +1
    }
  }
  if( (x$mean[loopNO] + (2*stdDevNO) >= h$meanNO2[loopNO]) && (x$mean[loopNO] - (2*stdDevNO) <= h$meanNO2[loopNO]) )
  {
    countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
```

```
flagOutlierTemp <- function(x,h)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)

 for(loop in 1:numLoop)
 {
  if(is.na(x$mean[loop]) || is.na(h$meantemp[loop]))
  {
   next
  }
  if(x$mean[loop] + (2*stdDevTemp) < h$meantemp[loop] || x$mean[loop] - (2*stdDevTemp) > h$meantemp[loop])
  {
   countFN <- countFN +1
  }
  if(x$mean[loop] + (2*stdDevTemp) >= h$meantemp[loop] && x$mean[loop] - (2*stdDevTemp) <= h$meantemp[loop])
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierTempWithCheckNeighbor <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  if(is.na(x$mean[loop]) || is.na(h$meantemp[loop]))
  {
   next
  }
  if(x$mean[loop] + stdDevTemp < h$meantemp[loop] || x$mean[loop] - stdDevTemp > h$meantemp[loop])
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
   for(loopNeighbor in 1: numNeighbor)
   {
    if(is.na(h$meantemp[loop]) || is.na(y[[loopNeighbor]]$meantemp[loop]))
    {
     next
    }
    if(y[[loopNeighbor]]$meantemp[loop] + stdDevNeighbor[loopNeighbor,1] <  h$meantemp[loop] || y[[loopNeighbor]]$meantemp[loop] -
              stdDevNeighbor[loopNeighbor,1] > h$meantemp[loop] )
    {
     flaggedOutlier <- flaggedOutlier + 1
    }
   }
   #only flagged if the value out of the range of all neighbor
   if(flaggedOutlier == numNeighbor)
   {
    countFN <- countFN +1
   } else {
    #flagged as TP
    countTP <- countTP +1
   }
  }
  if(x$mean[loop] + stdDevTemp >= h$meantemp[loop] && x$mean[loop] - stdDevTemp <= h$meantemp[loop])
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbor
flagOutlierTempWithCheckNeighborUsingOwnSD <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)
```

```
 for(loop in 1:numLoop)
 {
  if(is.na(x$mean[loop]) || is.na(h$meantemp[loop]))
  {
   next
  }
  if(x$mean[loop] + stdDevTemp < h$meantemp[loop] || x$mean[loop] - stdDevTemp > h$meantemp[loop])
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
   for(loopNeighbor in 1: numNeighbor)
   {
    if(x$mean[loop] + stdDevNeighbor < y[[loopNeighbor]]$meantemp[loop]  || x$mean[loop] - stdDevNeighbor > y[[loopNeighbor]]$meantemp[loop])
    {
     flaggedOutlier <- flaggedOutlier + 1
    }
   }
   #only flagged if the value out of the range of all neighbor
   if(flaggedOutlier == numNeighbor)
   {
    countFN <- countFN +1
   } else {
    #flagged as TP
    countTP <- countTP +1
   }
  }
  if(x$mean[loop] + stdDevTemp >= h$meantemp[loop] && x$mean[loop] - stdDevTemp <= h$meantemp[loop])
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
 return(count)
}

#x is ARIMA prediction
#h is next slot to be verified
#y is list of neighbors' arima model
flagOutlierTempWithCheckNeighborUsingOwnModel <- function(x,h,y)
{
 countFN <- 0
 countTP <- 0
 numLoop <- length(x$mean)
 #count the number of neighbor and compare with all of them
 numNeighbor <- length(y)

 for(loop in 1:numLoop)
 {
  if(is.na(x$mean[loop]) || is.na(h$meantemp[loop]))
  {
   next
  }
  if(x$mean[loop] + (2*stdDevTemp) < h$meantemp[loop] || x$mean[loop] - (2*stdDevTemp) > h$meantemp[loop])
  {
   #it's about to be flagged, but first check the neighbor
   #flagged only if the value is over all the neighbor's value
   flaggedOutlier <- 0
   for(loopNeighbor in 1: numNeighbor)
   {
    #skip if different length
    if(is.null(h$meantemp[loop]) || is.null(y[[loopNeighbor]]$mean[loop]))
    {
     next
    }
    if(y[[loopNeighbor]]$mean[loop] + (2*stdDevNeighbor[loopNeighbor,1]) < h$meantemp[loop]  || y[[loopNeighbor]]$mean[loop] -
              (2*stdDevNeighbor[loopNeighbor,1]) > h$meantemp[loop])
    {
     flaggedOutlier <- flaggedOutlier + 1
    }
   }
   #only flagged if the value out of the range of all neighbor
   if(flaggedOutlier == numNeighbor)
   {
    countFN <- countFN +1
   } else {
    #flagged as TP
    countTP <- countTP +1
   }
  }
  if(x$mean[loop] + (2*stdDevTemp) >= h$meantemp[loop] && x$mean[loop] - (2*stdDevTemp) <= h$meantemp[loop])
  {
   countTP <- countTP +1
  }
 }
 count <- data.frame(TP = countTP, FN = countFN)
```

```
 return(count)
}

#averaging all gasses field on x
summarizeData <- function(x)
{
 x$temperature.degC. <- as.numeric(x$temperature.degC.)
 x$humidity... <- as.numeric(x$humidity...)
 x$no2.ppb. <- as.numeric(x$no2.ppb.)
 x$co.ppm. <- as.numeric(x$co.ppm.)
 x$dd <- strptime(x$timestamp, format="%m/%d/%Y %H:%M:%S")
 #breaks data into hourly
 x$dfactor <- cut(x$dd, breaks="1 hour")

 dataSummary <- x[,-11] %>% group_by(dfactor) %>% summarise(meantemp=mean(na.omit(temperature.degC.)), meanNO2 = mean(na.omit(no2.ppb.)), meanCO =
                mean(na.omit(co.ppm.)),meanHumidity = mean(na.omit(humidity...)),n())
 return(dataSummary)
}

##a faster way to add list
##ref: http://stackoverflow.com/questions/17046336/here-we-go-again-append-an-element-to-a-list-in-r
Counter <- 0
Result <- list(NULL)
Size <- 1

AddItemDoubling <- function(item)
{
 if( .GlobalEnv$Counter == .GlobalEnv$Size )
 {
  length(.GlobalEnv$Result) <- .GlobalEnv$Size <- .GlobalEnv$Size * 2
 }

 .GlobalEnv$Counter <- .GlobalEnv$Counter + 1

 .GlobalEnv$Result[[.GlobalEnv$Counter]] <- item
}

## ADJUSTMENT FUNCTIONS
#f is formula
#x is data
runLR <- function(f, x)
{
 calLM <- lm(f, data=x)

 r2 <- summary(calLM)$r.squared
 intercept <- summary(calLM)$coefficients[1]
 predicted <- summary(calLM)$coefficients[2]
 interceptStdErr <- summary(calLM)$coefficients[3]
 predictedStdErr <- summary(calLM)$coefficients[4]

 result <- c(r2, intercept, interceptStdErr, predicted, predictedStdErr, length(x[,2]))
 result
}

#calculate root mean square error
#x is expected outcome, y is predicted values
rmse <- function(x, y)
{
 error <- x - y
 sqrt(mean(error^2, na.rm=TRUE))
}

#function to calculate r-square
#x is expected outcome, y is predicted values
coefficientOfDetermination <- function(x,y)
{
 yMean <- mean(x, na.rm = TRUE)
 numerator <- sum((y-yMean)^2, na.rm=TRUE)
 denominator <- sum((x-yMean)^2, na.rm=TRUE)
 numerator / denominator
}

#Implementation function for Multi Layer Perceptron
#x is data
#inputParams is input parameters in vector
#outputParam is output paramenter
#percenOfTraining is percentage of training
#hiddenActFunc is hidden activation function. Options: Act_Logistic
#outActFunc is output activation function. Options: Act_Logistic
#learnParameter is learning parameter
#maxiteration is number of loop or epoch
#numOfHiddenLayer is number of hidden layers
trainingMLP <- function(x, inputParams, outputParam, percenOfTraining, hiddenActFunc, outActFunc, learnParameter, maxiteration, numOfHiddenLayer)
{
 xLength <- length(x[,1])
 trainingLength <- round(xLength*percenOfTraining)
```

```
 dataMLP <-
                list(inputsTrain=x[1:trainingLength,inputParams],targetsTrain=x[1:trainingLength,outputParam],inputsTest=x[(trainingLength+1):xLength,inputParams],targe
                tsTest=x[(trainingLength+1):xLength,outputParam])

 netResult <- mlp(dataMLP$inputsTrain , dataMLP$targetsTrain, learnFunc = "Std_Backpropagation", learnFuncParams = c(as.numeric(learnParameter)), hiddenActFunc =
                hiddenActFunc, outputActFunc = outActFunc, maxit = maxiteration, size=numOfHiddenLayer)
 trainingPrediction <- predict(netResult, dataMLP$inputsTest)

 #construct both training and test into data frame
 trainingResult <- data.frame(target=dataMLP$targetsTrain,predict=netResult$fitted.values)
 testResult <- data.frame(target=dataMLP$targetsTest,predict=trainingPrediction)

 #add 1 on the formula indicating R to add regression line
 #for complete answer: www.ats.ucla.edu/stat/mult_pkg/faq/general/noconstant.htm
 lmTraining <- lm(target ~ 1 + predict, trainingResult)
 lmTest <- lm(target ~ 1 + predict, testResult)
 trainingRSquared <- round(summary(lmTraining)$r.squared,2)
 tesRSquared <- round(summary(lmTest)$r.squared,2)
 testRmse <- round(rmse(dataMLP$targetsTest, trainingPrediction),2)
 dVal <- round(dValue(dataMLP$targetsTest, trainingPrediction),2)

 result <- data.frame(trainRSquare = trainingRSquared, tesRsquare=tesRSquared, rmse=testRmse, d=dVal)
 result
}

#implementation function for Single Hidden Layer
#x is data
#inputParams is input parameters in vector
#outputParam is output paramenter
#percenOfTraining is percentage of training
#outActFunc is output activation function. Options: Act_Logistic
#learnParameter is learning parameter
#maxiteration is number of loop or epoch
#numOfHiddenLayer is number of hidden layers
trainingNNET <- function(x, inputParams, outputParam, percenOfTraining, outActFunc, learnParameter, maxiteration, numOfHiddenLayer)
{
 xLength <- length(x[,1])
 trainingLength <- round(xLength*percenOfTraining)
 #testLength <- xLength - trainingLength

 dataMLP <-
                list(inputsTrain=x[1:trainingLength,inputParams],targetsTrain=x[1:trainingLength,outputParam],inputsTest=x[(trainingLength+1):xLength,inputParams],targe
                tsTest=x[(trainingLength+1):xLength,outputParam])

 netResult <- nnet(dataMLP$inputsTrain , dataMLP$targetsTrain, size=numOfHiddenLayer, maxit=maxiteration, decay=learnParameter)
 if(!is.null(outActFunc))
 {
  trainingPrediction <- nnetPredInt(netResult, dataMLP$inputsTrain, dataMLP$targetsTrain, dataMLP$inputsTest, funName = outActFunc)
 } else {
  trainingPrediction <- predict(netResult, dataMLP$inputsTest)
 }

 #r squared is a statistic that will give some information about the goodness of fit of a model (wikipedia)
 #f squared coefficient is a statisctical measure of how well the regression line approximates the real data points (wikipedia)
 #therefore, to calculate r squared, we use lm function
 #construct both training and test into data frame
 trainingResult <- data.frame(target=dataMLP$targetsTrain,predict=netResult$fitted.values)

 if(!is.null(outActFunc))
 {
  testResult <- data.frame(target=dataMLP$targetsTest,predict=trainingPrediction$yPredValue)
 } else
 {
  testResult <- data.frame(target=dataMLP$targetsTest,predict=trainingPrediction)
 }

 #add 1 on the formula indicating R to add regression line
 #for complete answer: www.ats.ucla.edu/stat/mult_pkg/faq/general/noconstant.htm
 lmTraining <- lm(target~ 1 + predict, trainingResult)
 lmTest <- lm(target~ 1 + predict, testResult)
 trainingRSquared <- summary(lmTraining)$r.squared
 tesRSquared <- summary(lmTest)$r.squared

 testRmse <- rmse(dataMLP$targetsTest, trainingPrediction)
 dVal <-dValue(dataMLP$targetsTest, trainingPrediction)
 result <- data.frame(trainRSquare = trainingRSquared, tesRsquare=tesRSquared, rmse=testRmse, d=dVal)
 result
}

#x is the target data/expected outcome
#y is the predicted data
dValue <- function(x, y)
{
 xMean <- mean(x)
 nominator <- sum((y-x)^2, na.rm = TRUE)
 denominator <- sum((abs((y-xMean))+abs((x-xMean)))^2,na.rm = TRUE)
 result <- 1- (nominator / denominator)
```

```
 result
}

# data normalization
normalizeDt <- function(x)
{
 xmax <- max(x)
 xmin <- min(x)
 diffVal <- xmax-xmin
 nominator <- x-xmin
 result <- 2 * (nominator/diffVal) - 1
 result
}

#return normalized data into its normal unit
denormalizeDt <- function(x, xmin, xmax)
{
 diffVal <- xmax-xmin
 firstTerm <- ((as.double(x) + 1)*diffVal) / 2
 result <- firstTerm + xmin
 result
}

#implementation function for gradient boosting
#f is formula
#x is explanatory variables
#y is response variable
#learnParameter is learning parameter
#maxiteration is number of iteration
#numOfTraining is a fraction of training
#numOfCV is number of cross validation
#bestiteration decides which method to be used among the iterations: OOB, test, cv
#runningANNwithBoosting <- function(x, inputParams, outputParam, learnParameter, maxiteration, numOfTraining, trainFraction, numOfCV, bestiteration)
runningANNwithBoosting <- function(f, x, inputParams, outputParam, learnParameter, maxiteration, numOfTraining, trainFraction, numOfCV, bestiteration, bagFrac)
{
 xLength <- length(x[,1])
 trainingLength <- round(xLength*numOfTraining)

 trainData <- x[1:trainingLength,]
 testData <- x[(trainingLength+1):xLength,]

 #interaction.depth=1 means an additive model
 #bag.fraction is not required as the data has been splitted before. set to 1 --> WRONG ASSUMPTION
 #modelNeuralNet <- gbm(f, data=trainData, distribution = "gaussian", n.trees = nTrees, interaction.depth = 1, bag.fraction = bagFrac, train.fraction = trainFraction,
                cv.folds = numOfCV, shrinkage=learnParameter)
 modelNeuralNet <- gbm(f, data=trainData, distribution = "gaussian", n.trees = 100, interaction.depth = 1, bag.fraction = bagFrac, train.fraction = trainFraction, cv.folds =
                numOfCV, shrinkage=learnParameter)

 best.iter <- gbm.perf(modelNeuralNet, method = bestiteration)

 predictedTrainingResult <- modelNeuralNet$fit

 predictTestResult <- predict(modelNeuralNet, testData[,inputParams], best.iter)

 trainingResult <- data.frame(target=trainData[,outputParam],predict=predictedTrainingResult)
 testResult <- data.frame(target=testData[,outputParam],predict=predictTestResult)

 lmTraining <- lm(target~ 1 + predict, trainingResult)
 lmTest <- lm(target~ 1 + predict, testResult)

 #statistical results
 trainingRSquared <- summary(lmTraining)$r.squared
 tesRSquared <- summary(lmTest)$r.squared
 testRmse <- rmse(testData[,outputParam], predictTestResult)
 dVal <-dValue(testData[,outputParam], predictTestResult)
 result <- data.frame(trainRSquare = trainingRSquared, tesRsquare=tesRSquared, rmse=testRmse, d=dVal)
 result
}
```